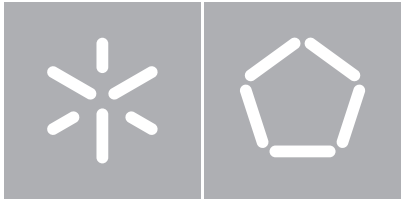
Universidade do Minho
Escola de Engenharia

Marco André Ferreira Reis

Desenvolvimento de um sistema integrado para o tratamento de dados de sequenciação de próxima geração

Outubro de 2013



Universidade do Minho
Escola de Engenharia
Departamento de Informática

Marco André Ferreira Reis

Desenvolvimento de um sistema integrado para o tratamento de dados de sequenciação de próxima geração

Dissertação de Mestrado
Mestrado em Bioinformática

Trabalho realizado sob orientação de

Miguel Francisco Almeida Pereira Rocha
Simão Pedro de Pinho Soares

Outubro de 2013

Agradecimentos

Antes de mais gostaria de agradecer ao professor Miguel Rocha e ao Eng. Simão Soares por todo o apoio dado durante a realização desta dissertação, mostrando-se sempre disponíveis a ajudar.

Quero também agradecer a todos os elementos da SilicoLife, nomeadamente o Paulo Vilaça, a Sónia Carneiro, o Hugo Costa, o João Cardoso e o Simão Soares, com que passei grande parte do tempo nos últimos meses, pelo conhecimento partilhado e pela amabilidade com que me receberam e integraram no grupo. Gostaria de destacar no entanto o Simão Soares, que me deu todas as condições necessárias para a realização deste trabalho, e o João Cardoso, que foi o meu "mestre" durante a realização do mesmo, e com quem aprendi muito.

Agradeço também ao professor Pedro Santos e ao Instituto de Higiene e Medicina Tropical de Lisboa que amavelmente disponibilizaram os dados sem os quais não teria sido possível apresentar os casos de estudo deste trabalho.

À minha mãe e aos meus avós, sem os quais não me teria sido possível frequentar o Mestrado de Bioinformática, nem me teria sido possível fazer este trabalho.

Por último, mas não menos importantes, aos meus colegas do Mestrado, pelos bons momentos que passei junto deles durante os últimos 2 anos. Gostaria no entanto de destacar o Pedro Barbosa, o Tiago Resende e a Ana Domingues, a quem eu agradeço em especial.

Gostava de agradecer também à Associação Universidade-Empresa para o Desenvolvimento - TecMinho pela bolsa de bolsa de investigação científica, no âmbito do projeto financiado ref. 014/TT/2013 – "Plataforma de tratamento de dados NGS".

Resumo

A sequenciação de próxima geração veio permitir a sequenciação em paralelo de milhões de pares de bases de *DNA* / *RNA*, tendo tido desde o início um grande impacto, ao ponto de se tornar o método escolhido em projetos de grande escala, em detrimento do método de Sanger. Entre as principais aplicações desta tecnologia encontram-se a análise em larga escala da metilação de *DNA*, o Chip-Seq para análise da interação entre proteínas e *DNA* ou *RNA*, e o mapeamento de rearranjos estruturais. Destacam-se, especialmente, a sequenciação de novos organismos ou indivíduos, o estudo de polimorfismos de nucleótido único (DNA-Seq) e a análise de expressão genética (RNA-Seq).

Neste trabalho, foi desenvolvido um sistema onde foram integradas ferramentas necessárias para estudos de DNA-Seq e RNA-Seq. Inicialmente, foi efetuado um estudo das aplicações existentes, tendo de seguida sido selecionadas as que se destacaram em parâmetros como a facilidade de utilização, documentação e possibilidade de integração com as restantes ferramentas do sistema. O sistema foi desenvolvido utilizando-se as linguagens de programação Ruby, Java e R, sendo as principais funcionalidades o estudo de polimorfismos, a assemblagem *de novo* e a análise de expressão genética a partir de dados de RNA-Seq. Este permite uma utilização simplificada e semiautomática dos vários programas, sendo acessível a utilizadores com poucos conhecimentos informáticos.

O sistema foi testado em três casos de estudo: caracterização de duas estirpes de *Mycobacterium Tuberculosis*, assemblagem *de novo* da *Pseudomonas* str. M1 e o estudo da expressão genética em amostras de *Saccharomyces cerevisiae*.

Abstract

Next-generation sequencing has enabled the sequencing of millions of base pairs of DNA and RNA, in parallel. This technology had, from the beginning a great impact to the point of becoming the method of choice for large-scale projects, replacing the Sanger method. Among the many applications of this technology we can include the analysis of DNA methylation, the analysis of the interaction between proteins (Chip-Seq) and DNA or RNA, and the mapping of structural rearrangements. However, the sequencing of new organisms or individuals, the study of single nucleotide polymorphisms (DNA-Seq) and gene expression analysis (RNA-Seq) are the main fields of study with this technology.

In this work, a system integrating tools to study DNA-Seq and RNA-Seq data has been developed, starting by studying existing applications. Then, taking into account parameters such as ease of use, documentation and possibility of integration with other system tools, an optimal set of tools has been selected.

The system was developed using the Ruby, Java and R programming languages, and its main features are the study of polymorphisms, *de novo* genomes assemblies and gene expression analysis. The developed system allows a simplified and semiautomatic use of the implemented tools making them accessible to users with limited computer knowledge.

The system was tested on three case studies: characterization of two strains of *Mycobacterium tuberculosis*, *de novo* assembly of *Pseudomonas* str. M1 and a study of gene expression in *Saccharomyces cerevisiae* samples.

Índice

Lista de Abreviaturas

BAM (Binary Alignment Map)

BWT (Burrows-Wheeler Transform)

cDNA (complementary DNA)

CDS (Coding DNA Sequences)

CSV (comma-separated values)

DNA (Deoxyribonucleic acid)

EC (Enzyme Commission)

GFF (Generic Feature Format)

GO (Gene Ontology)

GS20 (Genome Sequencer 20)

HGP (Human Genome Project)

HTML (HyperText Markup Language)

KEGG (Kyoto Encyclopedia of Genes and Genomes)

KO (Kegg Orthology)

mRNA (Messenger RNA)

NCBI (National Center for Biotechnology Information)

NGS (Next-Generation Sequencing)

OLC (Overlap Layout Consensus)

PCR (Polymerase Chain Reaction)

PE (Paired End)

RNA (Ribonucleic acid)

RPKM (Reads Per Kilobase per Milion mapped reads)

rRNA (Ribosomal RNA)

SAE (Small airway epithelium)

SAM (Sequence Alignment Map)

SE (Single End)

SFF (Standard Flowgram Format)

SGD (Saccharomyces Genome Database)

siRNA (small interfering RNA)

snRNA (small nuclear RNA)

SOLiD (Supported Oligo Ligation Detection)

SRA (Sequence Read Archive)

TC (Total count)

TMM (Trimmed Mean of M-values)

tRNA (Transfer RNA)

UQ (Upper Quartile)

VCF (Variant Call Format)

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

1.1 Contexto e motivação

O desenvolvimento de métodos de sequenciação alternativos ao método de Sanger [1], caracterizados pela possibilidade de sequenciar em paralelo milhões de pares de base de *DNA*, veio revolucionar a forma como os organismos são sequenciados. Estes métodos são identificados como métodos de sequenciação de próxima geração[1]. Como principais vantagens, quando comparados com os seus anteriores, destacam-se a velocidade com que os dados da sequenciação são gerados e o preço associado a esse processo [2], proporcionando o acesso a este tipo de tecnologia a pequenos e médios laboratórios.

Entre as principais aplicações desta tecnologia encontram-se a análise em larga escala de metilação de *DNA*, o ChIP-Seq[2], o mapeamento de rearranjos estruturais, a sequenciação de novos organismos e re-sequenciação genómica (*DNA-Seq*), a análise de expressão genética (*RNA-Seq*) [3] e o estudo de metagenomas. Na prática, esta tecnologia pode representar melhorias na área da saúde, onde o conceito de medicina personalizada lhe é constantemente associado [4], e também ao nível industrial, onde este tipo de sequenciação é bastante utilizado, como por exemplo na indústria alimentar [5].

Relacionadas com as vantagens da sequenciação de próxima geração, surgem as desvantagens ou limitações, nomeadamente a dificuldade inerente à análise e organização da grande quantidade de dados gerados. Devido a este fator, existe a necessidade de um grande poder computacional para a análise e gestão dos dados, não disponível em grande parte dos laboratórios ligados à biotecnologia, tal como a necessidade de profissionais com conhecimentos multidisciplinares, que façam a análise e interpretação dos dados de uma forma correta.

---

[1] Será usado o acrónimo anglo-saxónico de "*Next-Generation Sequencing*" (*NGS*), mais usado na comunidade.

[2] Técnica que combina a imunoprecipitação de cromatina com a sequenciação de próxima geração para análise da interação das proteínas com o DNA e RNA.

Dada a oportunidade, surgiram empresas que disponibilizam o serviço de análise, ou soluções informáticas comerciais acessíveis a profissionais com poucos conhecimentos informáticos, de onde se destacam a *Integromics* [6], a *Golden Helix* [7] e a *CLC bio* [8].

Por outro lado, existe uma grande quantidade de *software* livre, que permite cobrir todas as etapas na análise de dados de *NGS*. Como vantagem, a adoção deste tipo de recursos livres permite criar soluções adaptadas ao tipo e objetivo do projeto, havendo ainda a possibilidade de alteração e adaptação dos programas, nos casos onde os mesmos são de código aberto. A principal desvantagem, que é comum a muitas áreas de estudo da bioinformática, é a falta de normas e regras relativas aos tipos de ficheiros e procedimentos, e também a falta de documentação apropriada a utilizadores com poucos conhecimentos, tanto da área de informática como da biologia. Por esta razão, destacam-se os sítios *web* *SEQanswers* [9] e *Biostar* [10], como principal ponto de encontro entre utilizadores de ambas as áreas, com os autores dos programas, possibilitando dessa forma ter um melhor conhecimento do funcionamento dos mesmos. Em alguns casos, verifica-se mesmo um desenvolvimento das soluções orientado às necessidades expostas nos sítios *web*.

Este trabalho está enquadrado na dissertação de mestrado de Bioinformática da Universidade do Minho, e foi feito na empresa SilicoLife, Lda [11], companhia que se dedica à construção de soluções de Biologia Computacional para as Ciências da Vida, a partir da análise de informação de origem genómica, como por exemplo as áreas da biotecnologia industrial e saúde.

1.2 Objetivos

Tendo em conta os problemas identificados na secção 1.1, o objetivo global deste trabalho será o de criar um sistema integrado para o tratamento e análise de dados de sequenciação de próxima geração. Este sistema deverá permitir que, de uma forma semiautomática, seja feito o tratamento e análise de dados *NGS*, mais especificamente para estudos *DNA-Seq* onde existe um organismo de referência, *DNA-Seq* para novos organismos e *RNA-Seq*, utilizado na análise de expressão genética. Requisitos essenciais a este sistema são a necessidade do mesmo ser facilmente utilizável independentemente dos conhecimentos prévios do utilizador e de gerar documentação que auxilie na sua utilização.

Para o desenvolvimento do sistema foram identificadas tarefas para cada uma das funcionalidades necessárias:

- Alinhamento das leituras contra um genoma de referência**:** implementação de um sistema que permita fazer o controlo de qualidade, alinhamento, identificação de polimorfismos e

anotação dos mesmos, cálculo da sequência consenso e posterior identificação das *Coding DNA Sequences* (*CDS*).

- Montagem das leituras *de novo:* implementação de um sistema que permita fazer o controlo de qualidade, montagem das leituras *de novo* e aperfeiçoamento dos *contigs*[3] originados, utilizando ferramentas que permitam fazer o agrupamento dos mesmos em *scaffolds*[4], a eliminação de falhas existentes e a identificação das *CDS*.
- Análise da expressão genética**:** implementação de um sistema que permita fazer o controlo de qualidade, alinhamento, resumo, normalização e expressão diferencial das leituras. Deverá ainda permitir fazer a anotação funcional de um conjunto de genes de interesse, nomeadamente genes que apresentem diferenças significativas no que diz respeito à sua expressão em condições distintas.

O sistema final deverá permitir efetuar qualquer um dos estudos anteriormente referidos de uma forma simples e eficiente, tanto do ponto de vista de conhecimentos informáticos do utilizador, como do ponto de vista da definição de parâmetros. Este deverá ser, ainda assim, suficientemente flexível, para que um utilizador com conhecimentos avançados dos programas integrados possa utilizar os mesmos de uma forma relativamente familiar àquela como os utilizava anteriormente. O sistema será desenvolvido e implementado numa máquina com sistema operativo Linux, e para cada programa a integrar no sistema, será desenvolvido um módulo com vista a disponibilizar as funcionalidades dos mesmos, sendo para isso utilizadas várias linguagens de programação, nomeadamente Ruby, Java e R.

O objetivo final será a disponibilização de um sistema para a análise de dados *NGS*, e integração do mesmo com as restantes ferramentas desenvolvidas na SilicoLife, Lda.

1.3 Estrutura da tese

Este trabalho está dividido pelos seguintes capítulos:

- Na primeira parte da tese serão apresentados os resultados do levantamento bibliográfico feito na fase inicial, sendo apresentado o estado de arte no que diz respeito aos dados *NGS* e suas funcionalidades. O conjunto de ferramentas a utilizar é igualmente abordado, bem como os princípios de funcionamento das mesmas. (Capítulo 2)
- No Capítulo 3, relativo ao alinhamento contra um genoma de referência, é apresentada a forma como foi definida a *pipeline* e como esta foi implementada, tendo em conta os

[3] Sobreposição contínua de um conjunto de sequências.

[4] Resultado da união de vários *contigs* tendo em conta a ordem e orientação.

objetivos propostos. São ainda abordados de forma detalhada os programas selecionados para estudos de DNA-Seq com referência e um caso de estudo relativo à caracterização de duas estirpes de *Mycobacterium tuberculosis*.

- No Capítulo 4 serão referidos os programas selecionados para os estudos de DNA-Seq *de novo*, explicando de forma breve a razão pela qual foram selecionados. É ainda apresentado um caso de estudo referentes à assemblagem do genoma da *Pseudomonas* sp. M1 mostrando alguns dos comandos necessários para atingir os resultados apresentados.
- O Capitulo 5, tal como os capítulos anteriores, é constituído por uma secção direcionada à configuração do espaço de trabalho e apresentação dos programas selecionados e módulos desenvolvidos, e uma secção com um caso de estudo. O caso de estudo deste capítulo, referente ao RNA-Seq, é o estudo da expressão genética da *Saccharomyces cerevisiae* em duas condições de crescimento distintas.

Capítulo 2

Análise de dados de sequenciação de próxima geração

Neste capítulo, é explicado o que é a sequenciação de próxima geração, assim como os processos biológicos associados à mesma. É ainda feito o levantamento das tecnologias atualmente disponíveis e as suas vantagens e desvantagens, bem como os desafios que surgiram com o aparecimento deste tipo de dados.

2.1 Introdução

A sequenciação de Ácido Desoxirribonucleico (*DNA*) é um processo utilizado há bastante tempo, tendo sido publicado um dos procedimentos mais populares em 1977 [1], caracterizado pela sequenciação através do método de terminação da cadeia.

A sequenciação é o processo pelo qual a partir de uma amostra de *DNA* é gerado um arquivo eletrónico com os símbolos A, C, G, T, representativos das bases nitrogenadas Adenina, Citosina, Guanina e Timina. Este arquivo é composto por múltiplos símbolos, numa dada sequência, que representam as bases nitrogenadas contidas na amostra.

2.1.1 Conceitos de biologia molecular e celular

❖ *DNA*

Todas as células vivas armazenam informação hereditária sob a forma de moléculas de *DNA*. Estas moléculas são constituídas por duas longas cadeias em forma de espiral, compostas por quatro tipos de subunidades nucleotídicas, que codificam a informação genética (Figura 2-1 B).

As subunidades nucleotídicas, os nucleótidos, são formadas por um açúcar (desoxirribose), com um grupo de fosfato unido, e por uma base (Adenina, Guanina, Citosina ou Timina). Do ponto de vista estrutural, as bases Adenina e Timina, e Citosina e Guanina complementam-se, sendo este o conceito base da replicação de *DNA*. O *DNA* é sintetizado tendo por base uma cadeia de *DNA* preexistente, originando-se neste processo uma cadeia complementar, onde ocorre a ligação das bases complementares por ligações de hidrogénio. Estas

têm a particularidade de serem ligações fracas, permitindo dessa forma uma quebra fácil, simplificando a utilização das cadeias de *DNA* na replicação. As cadeias apresentam uma orientação singular, identificada por 3' quando a extremidade apresenta um hidroxilo, ou por 5' quando se trata um fosfato. Processos como a replicação ocorrem no sentido 5' -> 3', estando a ordem dos nucleótidos relacionada com a informação genética codificada.

A cadeia de *DNA*, que se trata de uma sequência de nucleótidos, encontra-se dividida em várias subsequências representativas de funcionalidades ou características específicas, às quais se dá o nome de genes. É composta ainda por subsequências cuja função é ainda incerta e por subsequências com função regulatória. Ao conjunto de todos os segmentos que constituem a cadeia de *DNA*, dá-se o nome de genoma, e o modo como este se encontra organizado é diferente de organismo para organismo. Nos organismos procariotas, organismos mais simples, os genes estão contidos numa única molécula de DNA, geralmente circular. Nos organismos eucariotas o *DNA* nuclear encontra-se tipicamente dividido por vários cromossomas, cada um referente a uma molécula de *DNA*. Por exemplo, as bactérias (organismo procariota) têm apenas um cromossoma, em contrapartida, o genoma humano (organismo eucariota) é composto por 46 cromossomas (23 pares).

❖ *RNA*

O *RNA* é composto por quatro tipos de subunidades nucleotídicas, tal como o *DNA*. No entanto, é composto pelo açúcar ribose, em vez de desoxirribose, e a base Timina compõe o *RNA*, dando lugar à base Uracilo. Esta base, tal como a Timina, é complementar à Adenina, havendo no entanto casualmente formação de pares Guanina – Uracilo. O conjunto de todo o *RNA* é identificado como transcriptoma, e abrange vários tipos de *RNA*, de onde se destaca o *mRNA* (*RNAm mensageiro*), responsável pela síntese das proteínas, e que compõe entre 3 e 5% do *RNA* total de uma célula. Por outro lado, o *rRNA* (*RNA ribossomal*) é o tipo de *RNA* mais presente na célula, existindo ainda o *snRNA* (*small nuclear RNA*) que direciona o *splicing*, o *tRNA* (*RNA de transferência*) essencial na tradução e o *siRNA* (*small interfering RNA*) com funções na expressão do genes em organismos eucariotas.

❖ Proteínas

As proteínas são as moléculas mais complexas e funcionalmente mais sofisticadas, sendo constituídas por uma cadeia de aminoácidos numa sequência específica, ligados entre si por uma ligação peptídica. Os aminoácidos são moléculas orgânicas compostas por um grupo amino e por um grupo carboxil, sendo codificados por grupos de três nucleótidos de um gene, os codões. Uma vez que existem sessenta e quatro codões diferentes e apenas 20 aminoácidos, existe uma codificação redundante dos mesmos. Erros no processo de replica-

ção de *DNA* podem causar alterações na sequência dos genes, mais concretamente, alteração, inserção ou remoção de nucleótidos. Estas alterações têm o nome de mutações, estando ainda o nome indel associado à inserção ou remoção de nucleótidos. Quando uma mutação ocorre em mais de 1% da população é considerada um polimorfismo de nucleótido único ou simples. Os polimorfismos de nucleótido único podem ou não codificar o mesmo aminoácido, alterando a forma e função da proteína.

Relativamente ao tamanho das proteínas, geralmente são constituídas por um valor entre o 50 e 2000 aminoácidos, e têm como principal funcionalidade dirigir todos os processos químicos que ocorrem na célula (enzima), podendo igualmente desempenhar funções na manutenção de estruturas, na regulação da expressão genética, no transporte de moléculas ou agir como anticorpos ou toxinas. Uma vez que muitas das proteínas surgem de genomas que evoluíram a partir do mesmo ancestral comum, é particularmente interessante a comparação de novas proteínas, com proteínas catalogadas em bases de dados. Pela comparação de duas sequências de aminoácidos referentes a proteínas, é possível pressupor a funcionalidade de uma proteína não conhecida, através do nível de identidade entre ambas, uma vez que sequências idênticas denotam estruturas tridimensionais semelhantes, que por sua vez indiciam funcionalidade igual ou aproximada.

❖ Dogma Central da Biologia

O modo em como é expressa a informação genética contida nos genes é comum a todos os organismos, e é considerado o dogma central da biologia. Consiste num conjunto de procedimentos sequenciais que permite transformar o *DNA* em proteínas, sempre que a célula necessite (Figura 2-1 A).

Uma vez que o *DNA* não é utilizado diretamente na síntese de proteínas, é essencial transformar o mesmo em *RNA*, num processo chamado transcrição. Trata-se de um processo que apresenta algumas particularidades conforme o tipo de organismo, pelo facto de existirem frações dos genes nos organismos eucariotas que não codificam a proteína, chamadas intrões. São regiões de maior tamanho em comparação com as regiões codificantes do gene, os exões, havendo a necessidade de uma série de processos posteriores à transcrição, até que o *RNA* atinja a sua forma final.

A transcrição nos organismos procariontes é bastante linear, iniciando-se com a cópia de uma das cadeias de *DNA* referente a um gene, sob a forma de *RNA* complementar. Neste processo há a abertura inicial de uma pequena porção de *DNA*, onde são expostas bases de uma das cadeias, que servirá de molde à síntese de *RNA*, originando-se neste processo uma cadeia de *RNA* complementar, por ação da enzima *rna*-polimerase (Figura 2-1 C). No caso dos

organismos eucariotas, o *DNA* é processado de igual modo, no entanto existem três tipos de *rna*-polimerase. A *rna*-polimerase II é responsável pela transcrição dos genes em pré-*mRNA*, estando a *rna*-polimerase I e a III responsáveis pela transcrição do *rRNA* e *tRNA* respetivamente.

Como já referido, o *RNA* resultante da transcrição do *DNA*, nos organismos eucariotas, necessita de uma fase de maturação (Figura 2-1 F) até que seja possível transformar os mesmo em proteínas. Nesta fase, as extremidades da molécula de *RNA* são alteradas, com a poliadenilação da extremidade 3 ' (ligação de caudas poli-A) e com o revestimento *cap* da extremidade 5', e os intrões são removidos. O processo pelo qual os intrões são removidos por ação de duas reações sequenciais de transferência de fosforil, onde há o corte e junção dos exões adjacentes a cada intrão é denominado *splicing*. O *splicing* está relacionado com o aparecimento de novas proteínas devido a um fenómeno de *splicing* alternativo (Figura 2-1 E), sendo ainda responsável pelo elevado potencial codificante dos genomas eucariotas. Este permite que a partir de uma dada sequência de pré-*mRNA*, sejam originadas diferentes moléculas de *mRNA*, que por sua vez sintetizam isoformas de proteínas distintas. Tal como a transcrição, os locais onde o *splicing* é efetuado é sinalizado por sequências especificas, equivalentes às sequências promotoras e de terminação do *DNA*.

Na tradução, os codões do *mRNA* são traduzidos para aminoácidos, ocorrendo a síntese de proteínas. A célula tem a capacidade de regular a produção de proteínas de acordo com as necessidades, sendo este controlo feito maioritariamente na produção de *mRNA* (Figura 2-1 D). [12]

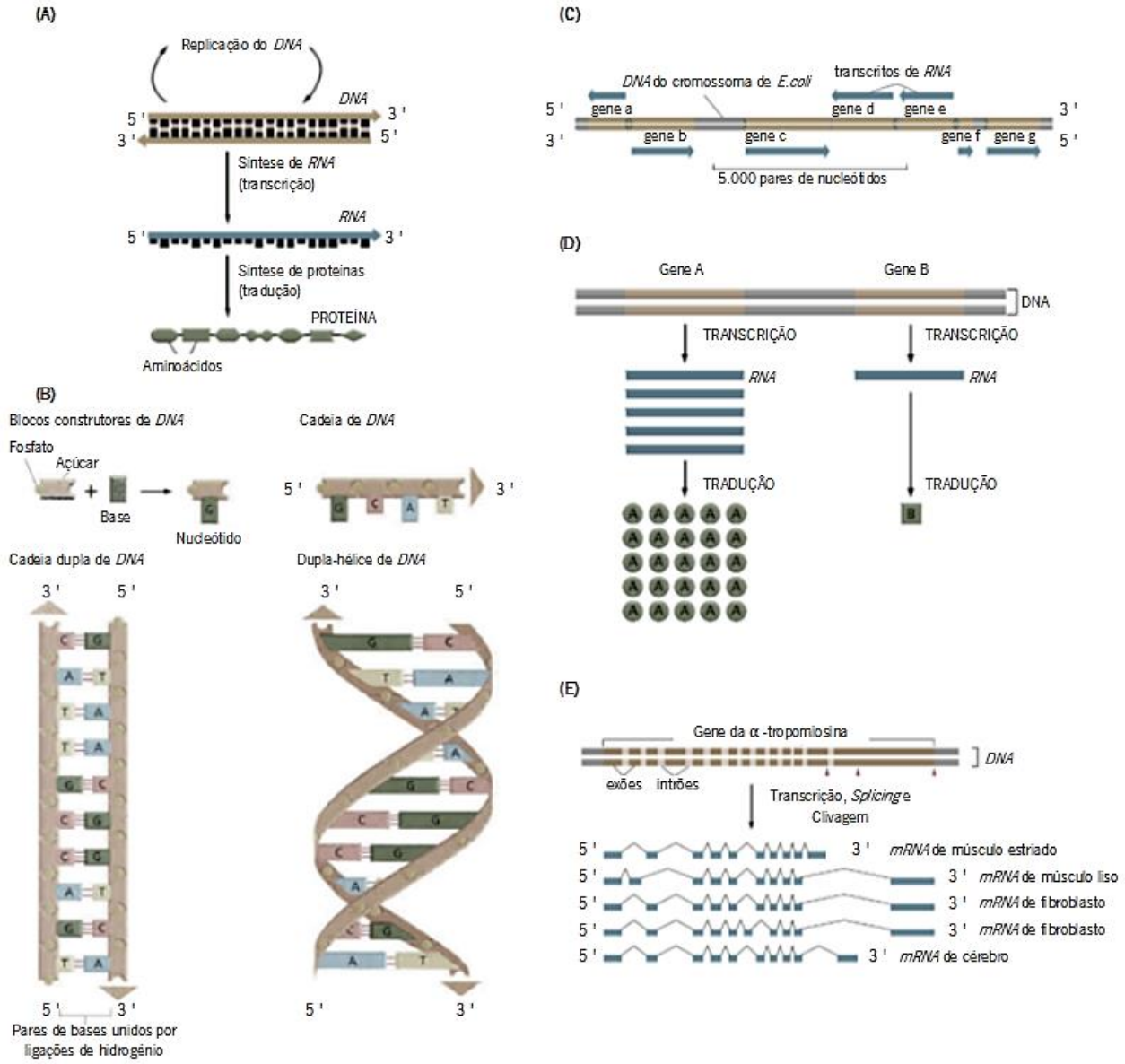


Figura 2-1 Conceitos de biologia molecular e celular. (A) Dogma central da biologia (B) Estrutura de uma molécula de DNA (C) Transcrição em ambas as cadeias de DNA, sendo o sentido determinado pelo promotor de cada gene (D) Diferenças na expressão genética entre diferentes genes. A expressão de um dado gene poderá ser igualmente diferente dependendo da necessidade (E) Splicing alternativo de um gene, onde são originados diferentes proteínas dependendo do local onde ocorre (F) Visão global do processo de expressão genética numa célula eucariota e procariota, onde é possível verificar a maior complexidade deste processo nos organismos eucariotas. Imagem adaptada de

2.1.2 Fase pré-sequenciação

A amostra é inicialmente recolhida e purificada, sendo necessária uma quantidade variável, dependendo da tecnologia de sequenciação e do protocolo de preparação adotado. O protocolo mais popular, e que serve de base aos restantes, é constituído por sete etapas:

1ª Corte ou fragmentação: o *DNA* da amostra é fragmentado em segmentos numa gama de tamanhos inferior a 800 pb (Illumina), num processo feito por ruptura mecânica ou por fragmentação enzimática.

2ª Reparação: as extremidades dos segmentos que foram danificados no processo de fragmentação são reparadas, numa reação enzimática onde se dá a remoção e preenchimento das bases danificadas.

3ª Limpeza e adição de Adenina: as enzimas utilizadas anteriormente são removidas, e é adicionada uma Adenina à extremidade 3', para impedir a junção dos segmentos e facilitar a ligação do adaptador.

4ª Ligação dos adaptadores: os adaptadores (pequenas sequências de *DNA* com uma função específica) são ligados a ambas as extremidades dos segmentos, tendo sempre em conta que um excesso de adaptadores poderá contaminar a amostra.

5ª Seleção de segmentos: os segmentos são selecionados tendo em conta o tamanho (geralmente entre os 300 e os 600 pb para a Illumina). São eliminados os segmentos que se ligaram a outros segmentos e filtrados segmentos dentro de uma gama de valores compatível com a máquina de sequenciação, sendo geralmente utilizado gel de agarose para este processo.

6ª Amplificação: a quantidade de *DNA* que contém adaptadores em ambas as extremidades é amplificado e são adicionadas sequências adicionais aos adaptadores para a hibridização.

7ª Controlo de qualidade: o último passo é verificar a biblioteca de *DNA* construída, verificando-se a concentração, o tamanho dos segmentos e se a amostra foi contaminada por algum produto não desejado.

Este processo irá determinar o tamanho das leituras, o tipo das leituras (*SE* ou *PE*[5]) e a distância entre as leituras, no caso de serem *PE*, sendo assim necessário utilizar um protoco-

[5] As leituras *PE* (*paired end*) são referentes ao facto de serem sequenciados ambos os extremos dos segmentos, dando origem dessa forma a duas leituras por segmento, sendo ainda guardada informação relativa à distância entre ambas. Esta distância é particularmente útil em casos de alinhamentos onde existem grandes re-arranjos estruturais e em zonas repetidas do genoma, sendo esta a principal vantagem em relação às *SE* (*single end*).

lo recomendado pelo fabricante da máquina em que se pretende sequenciar a amostra [13][14].

Em estudos onde são utilizadas amostras de *RNA* há a necessidade de efetuar procedimentos preparatórios, para que o protocolo acima descrito possa ser utilizado. Inicialmente, é eliminado todo o *DNA* da amostra, seguindo-se a filtragem de RNA mitocondrial, por exemplo com a utilização de seleção de caudas poli-A. Por ação da enzima transcriptase reversa, dá-se um fenómeno oposto à transcrição, formando-se cadeias de *cDNA* a partir do *RNA*. Dependendo da necessidade, podem ser utilizados protocolos que têm em conta a cadeia de origem do *RNA*, sendo particularmente interessante esta abordagem se houver a necessidade de descobrir novos genes e a orientação em que foi feita a transcrição. Por outro lado, os protocolos que não têm em conta esta situação, devido à sua maior simplicidade e menor custo, são os mais utilizados [15][16].

2.1.3 Tecnologias de sequenciação de próxima geração

Em 1998, inicia-se a revolução nos métodos de sequenciação de *DNA* com o aparecimento da ABI 3700, da Applied Biosystems [17]. Esta tecnologia permitia sequenciar num dia o que a maioria dos laboratórios demoraria um ano, e foi utilizada para sequenciar grande parte dos dados do *Human Genome Project* [18]. No ano de 2004, surge a primeira máquina de sequenciação de próxima geração comercializada, a GS20 da 454 Life Sciences Corporation [19]. Esta chamou a atenção da comunidade científica quando foi publicado o artigo de Margulies e colegas [20], onde foi apresentada a sequenciação e montagem do genoma da bactéria *Mycoplasma genitalium* com cobertura de 96% do genoma e uma precisão de 99,96%. O processo de sequenciação teve a duração aproximada de 14 horas, tendo sido feita apenas por um utilizador, e na altura significou uma grande evolução em relação aos métodos previamente utilizados.

Como consequência do sucesso desta tecnologia, naturalmente surgiram alternativas, nomeadamente a Illumina Genome Analyzer da Illumina [21], e a SOLiD da Applied Biosystems. Estas apresentam métodos de sequenciação distintos entre si, que originam quantidades diferentes de leituras[6]. A existência de várias empresas tem conduzido à diminuição do preço de sequenciação de um genoma (Figura 2-2), que era estimado em cerca de $19.500.000 em 2004 para um genoma humano, sendo sensivelmente $5.700 o valor atual, aproximando-se da meta dos $1.000 [22].

[6] Leituras correspondem aos fragmentos de *DNA* complementar (*cDNA*) da amostra. Associado a cada tecnologia de sequenciação está o tamanho destas leituras e o tipo de erros que podem apresentar.

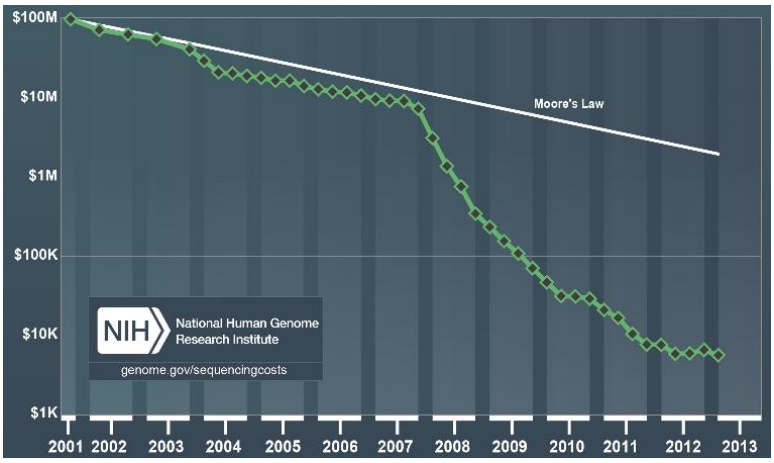


Figura 2-2 Evolução dos valores correspondentes ao custo de sequenciação de um genoma humano. Representados pela linha verde, estão os valores correspondentes a cada ano. Até ao aparecimento das máquinas de sequenciação de próxima geração, os valores observadores seguiam a lei de Moore. Imagem retirada de [23].

Neste contexto, as principais soluções tecnológicas são as seguintes:

- 454 Life Sciences**:** o princípio base de sequenciação da 454 é a pirosequenciação. O processo inicia-se pela fragmentação da amostra de *DNA*, seguindo-se a ligação individual dos fragmentos a *beads*[7]. Os segmentos presentes nas *beads* são amplificados por *PCR* (*Polymerase Chain Reaction*) e colocados no suporte de sequenciação. A partir da síntese de nucleótidos com os segmentos presentes nas *beads* dá-se a emissão de luz por ação da luciferase, permitindo associar a intensidade de luz ao número de nucleótidos incorporados (Figura 2-3). A sequência é o resultado deste processo, juntamente com informações relativas à posição de cada *bead* no suporte [24].
- illumina**:** a Illumina usa a sequenciação por síntese. Tal como na pirosequenciação, o *DNA* é fragmentado. Após a fase de fragmentação, são formados grupos, resultantes da amplificação dos fragmentos de *DNA*. Os nucleótidos do fragmento são calculados base a base, num processo em que são adicionados simultaneamente os quatro tipos de nucleótido contendo um identificador fluorescente (terminadores reversíveis), que com o auxílio da *DNA* polimerase, se vão anexar ao grupo (Figura 2-3). Todos os nucleótidos não incorporados são retirados e as bases anexadas no processo anterior são detetadas por uma câmara. De seguida, todos os terminadores são removidos por uma enzima, e o processo é repetido, até o fragmento estar completamente sequenciado [25].
- Applied Biosystems**:** distintamente em relação à 454 e à Illumina, que usam a síntese por *DNA* polimerase para fazer a sequenciação, a Applied Biosystems usa um método chamado de sequenciação por ligação. O modo de preparação da amostra é semelhante ao

[7] Esferas de captura de *DNA* onde são ligados os fragmentos compostos por pares de bases.

usado na 454, nomeadamente na ligação dos fragmentos a *beads* e sua amplificação. O sistema de ligação da ABI usa dezasseis identificadores fluorescentes, cada um deles contendo duas bases específicas, seguidas de três bases de clivagem (Figura 2-3). O sinal fluorescente é identificado no momento da ligação e os identificadores são clivados. A sequência do fragmento é calculada após cinco rondas de ligação [24].

- Outras: menos conhecidas, por serem mais recentes, existem a Ion Torrent (sequenciação por semicondutores)[26], a Pacbio (sequenciação em tempo real)[27] e a Nanopore (sequenciação usando nanoporos)[28]. Tanto a plataforma de sequenciação da Pacbio, como da Nanopore são referidas como máquinas de 3ª geração.

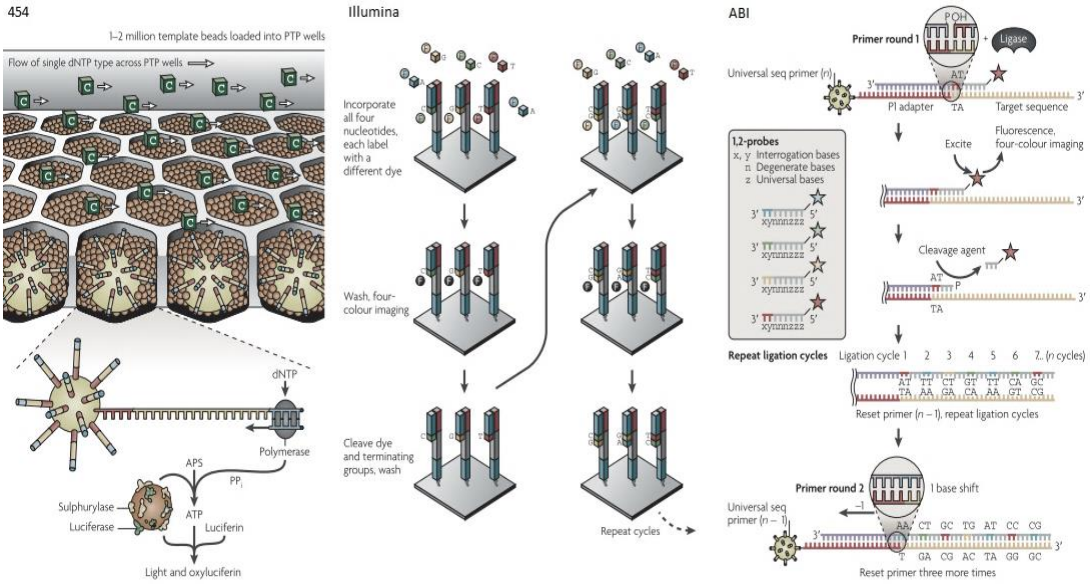


Figura 2-3 Métodos de sequenciação da 454 Life Sciences, Illumina e ABI. Imagem adaptada de [25].

Comparando as diferentes tecnologias, verifica-se que as mesmas se complementam, sendo usadas para diferentes aplicações. Por exemplo, a 454 GS, devido ao tamanho das leituras, é indicada para sequenciação de novos organismos, enquanto a SOLiDv4 é mais indicada quando é pretendido estudar os polimorfismos nos nucleótidos, devido à sua precisão. As principais diferenças encontram-se no tamanho das leituras, na quantidade de dados geradas pelas mesmas e no tempo que demoram a fazer a sequenciação. Na Tabela 2-1 são apresentadas as principais características de cada uma das máquinas, nomeadamente da 454 GS FLX, a HiSeq 2000 da Illumina e a SoliDv4 da ABI. Para efeitos de comparação, encontra-se também referida a Sanger 3730xl, que faz sequenciação através do método de Sanger.

Tabela 2-1 Características das máquinas de sequenciação de próxima geração mais utilizadas. Destaca-se o tamanho das leituras sequenciadas pela 454, e o rendimento da HiSeq, razão pela qual é a que apresenta valores mais baixos no preço por milhão de bases sequenciado. Por outro lado a SOLiD, apresenta uma precisão muito próxima do método de Sanger Tabela adaptada de [2].

	454 GS FLX	HiSeq 2000	SOLiDv4	Sanger 3730xl
Mecanismo Sequenciação	Pirosequenciação	Sequenciação por síntese	Sequenciação por ligação	Terminação de cadeia
Tamanho das Leituras	700 pb	50SE, 50PE, 100PE	50 + 35 pb ou 50 + 50pb	400 a 900 pb
Precisão das Leituras	99.9%	98%	99.94%	99.999%
Dados/funcionamento	0.7 Gb	600 Gb	120 Gb	1.9 a 84 Kb
Tempo/funcionamento	24 Horas	3 a 10 dias	7 dias para *SE* e 14 dias para *PE*	20 Minutos a 3 Horas
Vantagens	Tamanho das leituras, rápido	Alto rendimento	Precisão	Leituras longas de elevada qualidade
Desvantagens	Erros quando há mais de 6 bases seguidas repetidas, preço elevado, baixo rendimento	Leituras pequenas	Leituras pequenas	Custo elevado e pouco rendimento
Preço da máquina	$500.000	$690.000	$495.000	$95.000
Custo/milhão de bases	$10	$0.07	$0.13	$2400

2.1.4 Representação dos dados de sequenciação de próxima geração

O formato universal para se representarem sequências biológicas é o formato *FASTA*. A informação de um ficheiro neste formato, está organizada numa série de cabeçalhos, iniciados pelo símbolo >, seguindo-se geralmente um identificador e um comentário acerca da sequência na mesma linha. As linhas imediatas dizem respeito à sequência, estando cada nucleótido representado por um carácter. O código oficial para a representação dos nucleótidos data de 1984 [29], contendo também normas para representar bases ambíguas (Tabela 2-2).

Tabela 2-2 Código para a representação do DNA.

A	Adenina	T	Timina	Y	C ou T	K	G ou T	D	A ou G ou T	N	qualquer base
C	Citosina	U	Uracilo	S	G ou C	M	A ou C	H	A ou C ou T	. ou -	*gap*
G	Guanina	R	A ou G	W	A ou T	B	C ou G ou T	V	A ou C ou G		

Este formato é usado para representar as sequências de referência, a sequência consenso do alinhamento, os *contigs* e os *scaffolds*. O formato utilizado para representar as leituras, o *FASTQ*, é um formato derivado do *FASTA*, onde são acrescentados valores de qualidade para cada base (Figura 2-4). Estes estão codificados com caracteres ASCII, e são referentes ao valor *Phred* [30], a probabilidade da base ter sido detetada corretamente, e é calculado pela seguinte fórmula:

$$Q = -10log_{10}P_{erro}$$

onde P é a probabilidade da base ter sido identificada de maneira errada. Para um valor *phred*, por exemplo, de 30, a possibilidade da base estar errada é de 1 em 1000.

```
@SRR231654.2 GW6DJ:4:27 length=52
CGGACGTCTGGATAACAGCAACGCAAGCACGATGTACTACGCTCTACTTCTT
+SRR231654.2 GW6DJ:4:27 length=52
33-44+++33/444/11124/4)))-4-4----+++++++++)))))))+/3
```

Figura 2-4 Leitura no formato fastq. Primeira e terceira linha são referentes à identificação da sequência. A segunda linha é a sequência propriamente dita, constituída pela bases. A quarta linha contém os valores de qualidade correspondentes a cada base.

Depois de alinhadas as leituras, o resultado é guardado no formato genérico, chamado *SAM*. Este formato foi criado para representar o resultado do alinhamento das leituras contra uma referência (Tabela 2-3 e Tabela 2-4). Este tem uma versão binária, chamada *BAM*, que se encontra numa forma compacta e indexada, sendo por isso mais eficiente no que diz respeito ao acesso dos dados e ao tamanho do ficheiro [31].

Tabela 2-3 Cabeçalhos do ficheiro SAM.

Código	Descrição
@HD	Início do arquivo SAM e versão
@SQ	Identificação das sequências de referência
@RG	Identifica conjuntos de leituras dentro do arquivo
@PG	Programas utilizados
@CO	Comentários

Tabela 2-4 Campos referentes a uma sequência no formato SAM. Cada linha do ficheiro SAM tem os campos apresentados na tabela, permitindo assim identificar cada uma das sequências.

Col.	Nome	Descrição
1	QNAME	Nome da leitura
2	FLAG	Bits informativos
3	RNAME	Nome da sequência de referência
4	POS	Posição mais à esquerda da leitura que alinha na referência
5	MAPQ	Qualidade do alinhamento
6	CIGAR	Código Cigar (M - *Match*, I - *Insertion*, D - *Deletion*)
7	MRNM	Referência da próxima leitura do par
8	MPOS	Posição da próxima leitura do par
9	TLEN	Tamanho do *template*
10	SEQ	Sequência da leitura
11	QUAL	Qualidade da leitura codifica (ASCII-33)
12	OPT	Campos opcionais

Os *SNPs* estão também eles associados a um tipo de ficheiro padrão. Ainda que não seja utilizado pela totalidade dos programas, é o formato recomendado, uma vez que permite a utilização de um conjunto de ferramentas de análise desenvolvidas exclusivamente para o mesmo. Este é identificado por *Variant Call Format* (*VCF*) [32], e contém um conjunto de informações essenciais, como por exemplo, o cromossoma, posição, base na sequência de referência e base na nova sequência e a qualidade, que permite estabelecer um dado grau de confiança em relação ao *SNP*.

2.1.5 Fases na análise de dados de sequenciação de próxima geração

Os estudos de *DNA-Seq* podem ser divididos em três fases distintas, designadamente a fase de sequenciação, de onde resultam as leituras, a fase onde é feito o controlo de qualidade e, finalmente, o alinhamento contra uma referência ou a montagem *de novo* das leituras, dependendo se existe ou não um organismo semelhante, àquele de que foram retiradas as amostras de *DNA*. A partir das leituras alinhadas, são retiradas informações de interesse, como por exemplo os polimorfismos de nucleótidos únicos, quando as leituras são alinhadas contra uma referência [33]. Caso seja feita uma montagem *de novo* das leituras, geralmente o objetivo final é chegar a uma sequência final com o mínimo de falhas, e fazer a anotação da mesma.

Em relação às análises de *RNA-Seq*, é proposto por Oshlack [34] um conjunto de etapas que permitem fazer a análise da expressão diferencial, a partir das leituras iniciais. Tal como nos estudos de *DNA-Seq*, há a sequenciação da amostra inicial e o alinhamento ou montagem das leituras. As fases exclusivas a este tipo de estudo são a sumarização, normalização, expressão diferencial e a anotação funcional dos genes expressos distintamente (Figura 2-5).

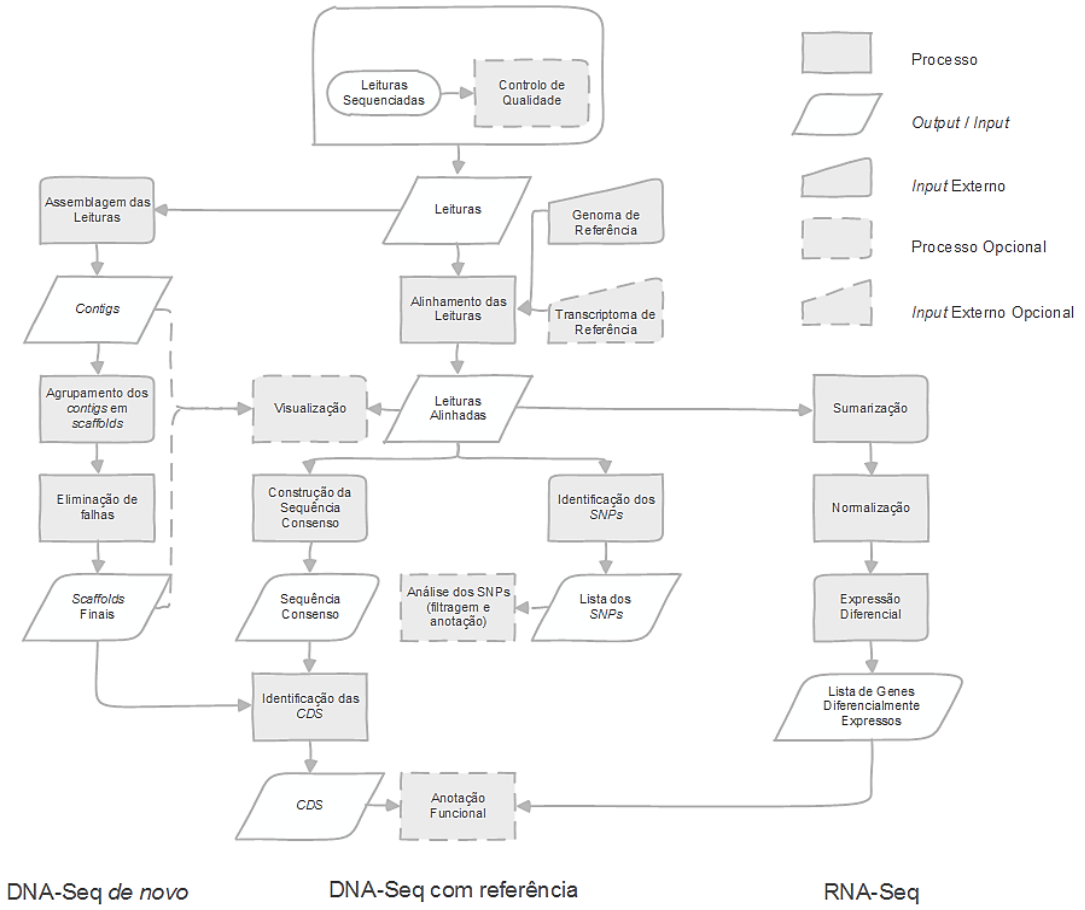


Figura 2-5 Pipeline definida para cada uma das funcionalidades a implementar. É possível visualizar as etapas em comum, bem como os dados de entrada e saída.

2.2 Aplicações para a análise de dados *DNA-Seq* com referência

O *DNA-Seq* veio oferecer uma alternativa válida ao método de Sanger, no que se refere à sequenciação de organismos. Algumas das utilizações deste método, quando existe um organismo próximo, cuja sequência possa ser usada como referência, é o estudo da variação genética e a descodificação de novos genes [35].

Como referido no ponto 1.1, existem aplicações comerciais que permitem fazer a análise de dados *NGS*. No entanto, associado às mesmas, está um elevado custo de utilização e o facto de não disponibilizarem o código fonte, de forma a serem adaptadas a situações específicas. Por outro lado, existe uma grande quantidade de ferramentas livres, que, de uma forma estruturada, permitem implementar as funcionalidades das soluções pagas, com a vantagem de poderem ser ajustadas tendo em conta o tipo de estudo a fazer com os dados iniciais.

Em seguida serão referidos *softwares* livres, que englobem as fases mencionadas na secção 2.1.5, e ferramentas auxiliares que permitam a interpretação dos dados.

2.2.1 Controlo de qualidade

O primeiro passo após a sequenciação das amostras é fazer o controlo de qualidade das leituras. Trata-se de um passo facultativo, não mencionado em muitos dos artigos referentes a análises *NGS*. A origem digital dos dados, juntamente com o controlo de qualidade aplicado por cada fabricante, leva muitos investigadores a desconsiderar a necessidade deste passo. No entanto, verificam-se erros nas leituras associados a cada uma das plataformas, mesmo após o controlo de qualidade aplicado pelos fabricantes. Os erros mais comuns são a baixa qualidade das leituras, a contaminação pelos adaptadores e os *indels*[8] [36]. Estes erros podem conduzir a conclusões erradas, como por exemplo no estudo de polimorfismos.

O objetivo do controlo de qualidade é, portanto, fazer uma avaliação das leituras, geralmente com ferramentas de visualização indicadas para as mesmas e, de seguida, proceder-se à correção, remoção e corte das leituras que não respeitem os valores mínimos de qualidade pretendidos [37]. Alguns dos parâmetros a ter em conta são o tamanho das sequências, a qualidade de cada base e da leitura, dada pelos valores *phred*, o número de sequências repetidas e a distribuição das bases.

As primeiras ferramentas desenvolvidas para este fim foram a *TileQC* [38] e a *PIQA* [39], que tinham em comum o facto de serem indicadas apenas para leituras das plataformas da Illumina. Atualmente, para a visualização dos dados destaca-se o *FastQC* [40] e o *NGS QC Toolkit* [41]. Estas permitem fazer relatórios detalhados relativos aos vários parâmetros de qualidade.

Com a análise do relatório de qualidade, é possível a utilização de programas que permitam corrigir as leituras que se destacaram negativamente. Nesta categoria, destaca-se mais uma vez o *NGS QC Toolkit*, tal como o *FASTX-Toolkit* [42] e o *NGSQC* [36].

2.2.2 Alinhamento contra uma referência

Se existir um organismo suficientemente semelhante, que possa ser usado como referência, as leituras podem ser mapeadas tendo em conta a sequência desse organismo. Esta é a fase mais crítica na análise de dados, uma vez que é este resultado, que todas as outras operações vão processar. Há que ter em conta diferentes fatores, mais especificamente, a

[8] Refere-se a ocorrências de inserção ou deleção de nucleótidos, podendo provocar alterações no modo em como o genoma é traduzido, quando o tamanho do *indel* não é um múltiplo de 3.

máquina em que as leituras foram sequenciadas, o tamanho das leituras, a quantidade de leituras, se as leituras são *SE* ou *PE* e o genoma a utilizar como referência.

Métodos de alinhamento anteriormente usados, como o *BLAST* [43], não foram desenvolvidos tendo como objetivo a utilização de dados *NGS*. Para fazer o alinhamento de grandes quantidades de dados, onde é pretendido mapear pequenas leituras a um genoma de referência, foram aplicados novos métodos, que permitam resolver o problema corretamente em tempo útil. Estes resultam do aperfeiçoamento de métodos já existentes, como as tabelas de dispersão e o *seed-and-extend*, bem como de métodos desenvolvidos especialmente para este problema, como o *Burrows-Wheeler Transform* (*BWT*), derivado de árvores de sufixos. Estes métodos têm em comum o facto de serem heurísticos, como tal não garantem a solução ótima. Do ponto de vista computacional, o *BWT* é mais eficiente, quer em termos de memória, quer de processamento dos dados. Deste modo, trata-se do método mais utilizado atualmente [44] [45].

A maioria dos programas de alinhamento contra uma referência funciona num simples computador de secretária, dependendo no entanto da quantidade de leituras e do tamanho do genoma de referência. É um processo onde as leituras no formato *FASTQ* são mapeadas tendo em conta uma referência, dando origem a um ficheiro no formato *SAM*. Alguns *softwares*, como por exemplo o *SOAP2* [46], não respeitam esta norma, dando origem a um ficheiro com outro formato, havendo a necessidade posterior de o converter para o formato *SAM*,. Alguns deles apresentam funcionalidades, como por exemplo a possibilidade de inserção de falhas e o facto de permitirem computação paralela, que podem influenciar na altura da escolha do programa a utilizar.

Os programas existentes podem agrupar-se da seguinte forma:

- Programas baseados em tabelas de dispersão**:** algoritmos baseados em tabelas de dispersão permitem indexar dados complexos, possibilitando uma pesquisa eficaz dos mesmos. No caso de dados *NGS*, tanto as leituras como a sequência de referência podem ser indexados numa estrutura de dados deste tipo [45]. Nesta classe de programas, destacam-se o *MAQ* [47], um dos primeiros programas desenvolvidos para alinhar sequências *NGS*, o *GMAP* [48] e o *SSAHA2* [49].
- Programas baseados em *Burrows-Wheller Transform***:** o princípio dos algoritmos que usam *BWT* reside no facto de usarem como estrutura de dados o *FM índex*[9], que não é mais

[9] "*Full-text index in Minute space*", permite a compressão de texto numa estrutura de dados eficiente, que permite igualmente a pesquisa rápida de *substrings*.

que um *array* de sufixos construído não pela sequência original, mas pela sequência após lhe ter sido aplicado o *BWT*. As vantagens desta estrutura de dados são a facilidade de pesquisa, e a nível de compressão dos dados, que possibilita indexar o genoma humano em 2.3 *GB*. Comparando com os programas baseados em tabelas de dispersão, este grupo de programas é muito mais rápido, apresentando ainda assim um nível de sensibilidade idêntico. Nesta categoria destacam-se o *Bowtie* [50], o *Bowtie2* [51] e o *BWA* [52].

Tabela 2-5 Tabela com características dos programas de alinhamento contra uma referência. A coluna Plat.Seq refere-se à compatibilidade com os dados das máquinas de sequenciação de próxima geração (I-Illumina, So-ABI Solid, 4-454). As colunas Entrada e Saída são referentes aos ficheiros de entrada e saída. O S, na coluna Falhas, indica que o programa é compatível com a abertura de falhas. O Alinhamento pode ser G-Global, L-Local ou GL-Global e Local. As colunas Temp.-Tempo tolal do alinhamento, Mem-Memória utlizada e Leit.Alin-Leituras Alinhadas, permitem fazer uma comparação entre os vários programas. Tabela adaptada de [53].

Programa	Plat.Seq	Entrada	Saída	Falhas	Alinhamento	Paralelo	PE	Temp.	Mem	Leit.Alin	Cit./ano
Bowtie	I,So,4	FASTA/Q	SAM TSV	S	GL	S	S	169	5	798566	37
Bowtie2	I,4	FASTA/Q	SAM TSV	S	GL	S	S	176	5.1	991880	beta
Bwa	I,So,4	FASTA/Q	SAM	S	G	S	S	97	7.6	928000	224
GMAP	I,4	FASTA/Q	SAM GFF	S	GL	S	N	2887	7.6	998454	29
MAQ	I,So	FASTA/Q	TSV	S	-	N	S	-	-	-	251
SSAHA2	I,4	FASTA/Q	SAM	N	L	N	S	207	9.5	1,0E+06	45

Fonseca [53] fez um levantamento dos programas de alinhamento de sequências de *NGS*, e das suas características, procurando testar, utilizando os mesmos dados de entrada para todos os programas, o tempo de indexação dos dados, tempo de alinhamento, memória utilizada e número de leituras alinhadas. Os que mais se destacaram foram o *Bowtie2*, tendo em conta o tempo/memória utilizada por leituras alinhadas, e o *SSAHA2* se tivermos em conta apenas o número de leituras alinhadas (Tabela 2-5).

2.2.3 Identificação das sequências codificantes

Uma das hipóteses após o alinhamento é construir a sequência consenso em relação às leituras alinhadas. Esta sequência consenso permitirá posteriormente fazer a anotação do genoma. Para isso, através de um processo feito pelo programa *SAMtools* [31], o ficheiro *SAM*, resultante do alinhamento, é transformado num ficheiro *FASTQ*, que posteriormente pode facilmente ser convertido no formato *FASTA*. Este ficheiro possui as bases consenso, resultantes do mapeamento das leituras, e será do tamanho da sequência de referência.

Este é um processo de estruturação do genoma, retirando informações de interesse. Desse modo são calculadas as *CDS*, sendo o objetivo final fazer a anotação funcional das mesmas, ou seja, associar um gene e a sua função biológica a cada *CDS*. Para esta funcionalidade,

uma das soluções mais reconhecidas é o servidor de anotação *RAST* [54]. No caso de estudos *DNA-Seq* com referência, este processo tem uma importância relativa, uma vez que as leituras são alinhadas contra um genoma de referência que na maioria dos casos já se encontra anotado. Este processo, e as ferramenta utilizadas para o mesmo, será assim descrito mais detalhadamente na secção 2.3.5.

2.2.4 Polimorfismos de nucleótidos simples

Um polimorfismo de nucleótido simples, ou *SNP*, é a designação dada a uma alteração nas bases de *DNA* do genoma, e que afeta pelo menos 1% dos indivíduos de uma população. São considerados polimorfismos de nucleótido único quando há alteração de apenas uma base, e no caso de introdução ou eliminação de bases, são identificados por *indel*. Se afetar menos de 1% da população, estas alterações passam a ser consideradas mutações. Do ponto de vista biológico, os polimorfismos têm grande importância, uma vez que são responsáveis por grande parte da variabilidade genética entre os indivíduos de uma população.

Os polimorfismos ocorrem tanto em regiões codificantes (exões), como em regiões não codificantes ou intergénicas (intrões), e estão distribuídos por todo o genoma. Quando ocorrem nos exões, estas pequenas transformações podem originar alterações na resposta imunitária do organismo, sendo muitas vezes associadas à resposta a doenças, vírus e fármacos.

O processo de estudo dos polimorfismos é dividido por duas fases, nomeadamente a fase de cálculo dos mesmos, e uma fase de filtragem e anotação. Para o cálculo de *SNP*, do elevado número de soluções existentes, destacam-se o *SAMtools* [31], o *GATK* [55], o *SOAPsnp* [56] e o *Sniper* [57]. O cálculo dos *SNP* é feito a partir de métodos probabilísticos [58], havendo no entanto, grandes diferenças em relação ao número de *SNP* calculados entre os diferentes programas, como é mostrado por Pabinger *et al.* [37]. Estes programas originam um ficheiro *VCF* [59], com informações relativos aos polimorfismos, como por exemplo, cromossoma onde ocorreu, posição, base correspondente na sequência de referência e confiança no cálculo do polimorfismo.

O processo de filtragem geralmente acontece numa fase anterior à fase de anotação, e pode ser o processo pelo qual são eliminados polimorfismos calculados com baixo nível de confiança, ou filtrados polimorfismos de um dado cromossoma, por exemplo. O programa indicado para estas operações é o *SnpSift* [60]. Outra abordagem, particularmente interessante quando se estudam várias estirpes de um dado organismo, é fazer a comparação dos vários ficheiros *VCF* calculados, e selecionar os polimorfismos comuns às várias estirpes, ou

pelo contrário, filtrar os que são exclusivos a cada estirpe. Esta abordagem permite associar características diferenciadoras das estirpes, em relação a uma referência, aos polimorfismos comuns entre as mesmas. Um dos programas desenvolvidos exclusivamente para este efeito é o *VcfTools* [32].

Por último, de modo a retirar informações úteis do ponto de vista biológico, é necessário proceder-se à anotação dos vários polimorfismos. Trata-se de um procedimento que procura elucidar o impacto de cada polimorfismo, como por exemplo, se introduz um novo codão *stop* ou se ocorre numa região codificante do genoma. Dos programas analisados para este fim, destaca-se o *SnpEff* [61], pela quantidade de organismos a que oferece suporte e pela informação apresentada. Existem no entanto programas alternativos como o *ANNOVAR* [62] e o *Vep* [63].

2.2.5 Visualização dos dados

O ficheiro resultante do alinhamento é passível de ser visualizado. Pelo facto dos ficheiros resultantes apresentarem diferenças, conforme a aplicação usada para o alinhamento, os visualizadores para este tipo de dados deverão ser flexíveis de modo a permitir a correta observação dos dados. Estes devem ainda ser rápidos e eficazes ao ler grandes quantidades de dados (ficheiros *SAM)*, ou a ler arquivos no formato binário (ficheiros *BAM*) [64]. Estes permitem fazer um rastreio de possíveis problemas no alinhamento, e também possibilitam a visualização de zonas específicas do genoma, ver a qualidade das bases, a cobertura quanto ao número de leituras numa zona específica da sequência e fazer a validação dos *SNPs* (Figura 2-6).

Para a visualização de alinhamentos contra uma referência e de dados de *RNA-Seq*, discutido mais em detalhe na secção (2.4), evidenciam-se o *MapView* [65], o *Tablet* [66] e o *IGV* [67]. Com funcionalidades mais específicas para o estudos de montagem *de novo*, existe o *Mauve* [68], que permite a visualização dos vários *contigs* ou *scaffolds*.

Figura 2-6 Imagem de um alinhamento retirada do programa Tablet. Permite visualmente investigar a existência de SNPs, o número de leituras alinhadas e verificar individualmente as leituras.

2.3 Aplicações para a análise de dados *DNA-Seq de novo*

A assemblagem ou montagem *de novo* é o processo em que se constrói um genoma, sem que para isso, seja usada uma referência. É um procedimento bastante complexo e demorado, tanto em termos computacionais, como do ponto de vista do utilizador, nomeadamente na interpretação dos resultados e na definição dos passos necessários, de modo a que o genoma tenha o menor número possível de falhas e *scaffolds*. As leituras são comparadas e alinhadas entre si, permitindo desta forma montar segmentos de sequência cada vez maiores, até eventualmente se chegar a uma sequência consenso, correspondente ao genoma final [69].

Em termos de comparação com a análise de dados *DNA-Seq* com referência, poderá dizer-se que não há fases em comum, tirando o controlo de qualidade das leituras. Do ponto de vista do alinhamento, os programas utilizados, tal como os princípios de funcionamento dos mesmos são diferentes. No que diz respeito ao cálculo de *CDS*, as ferramentas e princípios são iguais, no entanto, geralmente quando o objetivo é fazer a anotação de um genoma, é feita uma assemblagem das leituras, sem utilização de qualquer referência.

2.3.1 Montagem *de novo*

Os métodos de alinhamento utlizados no alinhamento contra uma referência, não podem ser utlizados para resolver este tipo de problemas. Para este tipo de alinhamentos foram desenvolvidos novos métodos de alinhamento, podendo estes ser divididos em três categorias,

todas elas baseadas em grafos[10]. Essas categorias são os algoritmos *Greedy*, *OLC* (*Overlap Layout Consensus*) e os baseados em grafos de *de Bruijn* [70].

- Algoritmos Greedy**:** os algoritmos *Greedy* foram os primeiros a ser implementados. Estes consistiam em ir adicionando sucessivamente a leitura com maior *overlap*[11], em relação a uma leitura ou *contig* inicial. O principal defeito destes algoritmos era o facto de não garantirem uma solução ótima. Por exemplo, quando o algoritmo adiciona uma leitura que poderia ser utilizada posteriormente para originar um *contig* ainda maior. O *SSAKE* [71] foi o primeiro programa desenvolvido para a montagem deste tipo de leituras, e usava este algoritmo.
- *Overlap Layout Consensus* (*OLC*)**:** é um algoritmo otimizado para genomas grandes, e é um método geralmente utilizado para fazer a assemblagem de leituras originadas pelo método de Sanger. Inicialmente, é criado um grafo de *overlaps*, sendo a partir desse grafo descoberto um caminho que irá permitir reconstruir a sequência consenso. A construção do grafo de *overlaps* usa um algoritmo de *seed and extend*, para comparar todas as leituras entre si. Cada nó do grafo é referente a uma leitura, sendo atribuído um peso a cada aresta, correspondente ao *overlap* entre as strings de cada nó. O objetivo final é encontrar um caminho de *Hamilton*, caminho que visita todos os vértices uma única vez [72]. O software mais conhecido desta categoria é o *Newbler*, atualmente conhecido como *GS de novo assembler* [73], que é distribuído pela *454*, o *Edena* [74] e o *Celera* [75].
- Grafo de *de Bruijn***:** este algoritmo utiliza igualmente o conceito de grafo. Nesta abordagem as leituras são divididas em *k-mers*, que são *strings* de tamanho k, sendo cada aresta do grafo correspondente a um k-mer da sequência. Os nós do grafo são estruturas que representam *(k-1)-mers*, sendo a direção atribuída de um prefixo para um sufixo (Figura 2-7). A sequência consenso é calculada pelo caminho de *Euler*, ou seja, um caminho que inclua todos os *k-mers*, uma única vez [72]. Geralmente, é aplicado para fazer a montagem de leituras das máquinas da Illumina e da SOLiD, e é a base de programas como o *Velvet* [76], o *ABySS* [77] e o *SOAPdenovo* [78]. A forma como o grafo de *de Bruijn* é aplicado, depende de programa para programa, como é apresentado no artigo de Miller [70].

[10] Objectos básicos para representação de problemas da vida real. Um grafo é representado por pontos (vértices) e por rectas que ligam os vértices entre si, chamadas arestas.

[11] Tamanho da maior *substring* comum.

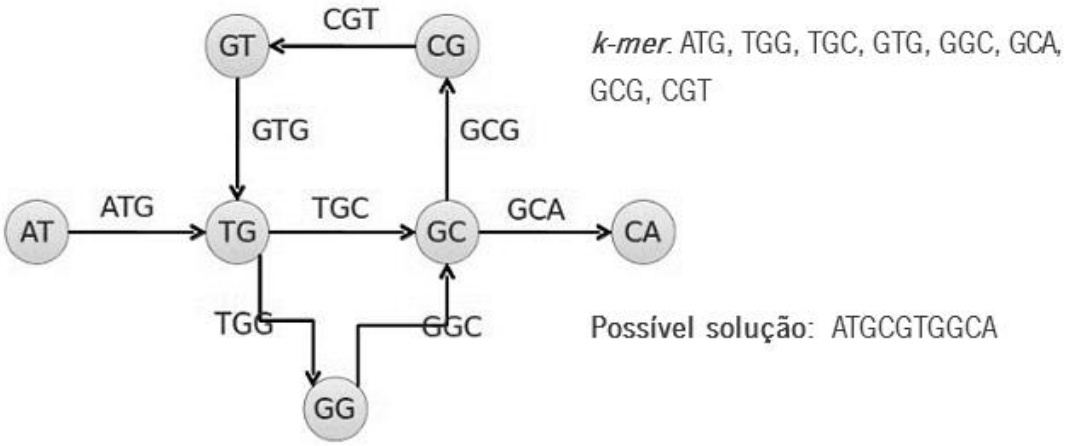


Figura 2-7 Representação de um grafo de de Bruijn. Grafo composto por k-mers de tamanho 3. Imagem adaptada de [72].

Os programas mais utilizados atualmente são o *ABySS*, o *Velvet*, e o *SOAPdenovo*, e têm em comum o facto de utilizarem leituras *PE*, para fazer a assemblagem. No estudo de Zhang [79], destacaram-se o *SOAPdenovo* em termos de velocidade e o *Taipan* [80], no que diz respeito à qualidade da assemblagem. No guia apresentado pela empresa Illumina é sugerida a utilização dos programas conforme a origem das leituras, e é recomendada a utilização do *Velvet* ou do *SOAPdenovo* para genomas de bactérias, e o *ABySS* para os restantes casos [81]. Quando utilizadas leituras provenientes da *454*, o *Newbler* destaca-se dos restantes, tendo como contrapartida o facto de ser pago. Em relação a assemblagens híbridas, onde são utilizadas leituras de diferentes tecnologias, são considerados soluções bastante eficazes o *Celera* e o *Mira* [82].

Apesar da melhoria a nível de recursos, eficácia e paralelização dos algoritmos, estes são geralmente aplicados apenas para fazer a assemblagem de genomas de bactérias, ou para construir pequenas regiões de *DNA* do genoma humano, devido ao elevado número de repetições encontradas no genoma humano [64].

2.3.2 Agrupamento dos *contigs* em *scaffolds*

O processo seguinte à montagem das leituras em *contigs* é o agrupamento dos mesmos em *scaffolds*. Usando informação relativa da distância entre leituras *PE* ou *mate-pair*[12], é possível obter a ordem, distância e orientação dos *contigs*. Este processo permite a redução significativa do número de *contigs*, dando origem a *supercontigs* ou *scaffolds* (Figura 2-8).

[12] Leituras idênticas às leituras PE, mas com distância superior entre as leituras (de 2000 a 5000 bases de distância).

A maioria dos programas de assemblagem de genomas permite efetuar este processo, como por exemplo o *ABySS* e o *SOAP*. No entanto, os parâmetros não podem ser determinados pelo utilizador. Um dos programas usados para construir os *scaffolds* quando utilizado o método de Sanger para sequenciação, é o *Bambus* [83]. Para dados de sequenciação de próxima geração, foram igualmente desenvolvidos *softwares* que permitem ao utilizador determinar os *scaffolds* independentemente do programa utilizado para construir os *contigs*, nomeadamente o *SOPRA* [84] e o *SSPACE* [85].

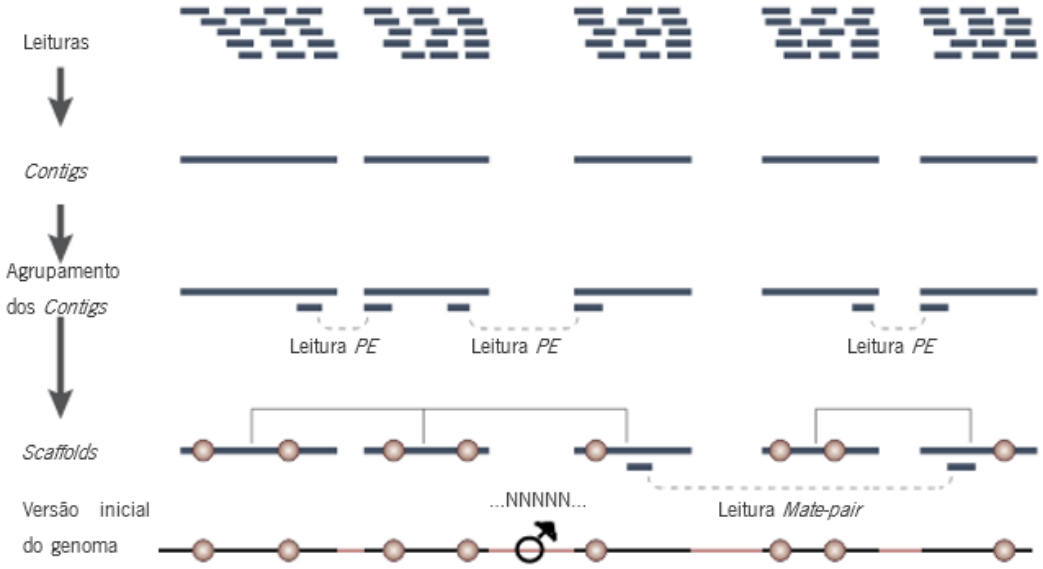


Figura 2-8 Fases de uma assemblagem de novo. Inicialmente, as leituras são alinhadas entre si, constituindo Contigs. Os contigs são agrupados, utilizando leituras PE, dando origem a contigs maiores, os scaffolds. Num processo iterativo, os scaffolds vão constituindo scaffolds cada vez maiores, com o auxílio de leituras PE e mate-pair. O resultado deste processo é o esboço do genoma final, uma vez que contem zonas não sequenciadas, identificadas pela base N. Imagem adaptada de [86].

2.3.3 Eliminação de falhas

Zonas de baixa cobertura e zonas repetidas continuam a ser o principal problema na altura de completar o genoma. Apesar do auxílio dado pela construção dos *scaffolds*, este não acrescenta nova informação ao genoma inicial. O processo de eliminação de falhas é de grande importância, uma vez que a eliminação de falhas manual através da sequenciação de Sanger é um processo bastante dispendioso [87].

Métodos baseados em grafos de *de Bruijn* foram implementados para este fim, por exemplo, o *GapCloser* [88], implementado pelos mesmos programadores do *SOAPdenovo* e o *IMAGE* [89]. O principal defeito na abordagem destes dois programas é o facto de não terem em conta o tamanho das falhas, calculadas anteriormente no processo de *scaffolding*. O programa GapFiller [87], desenvolvido por Boetzer, tem em conta as informações adquiri-

das no processo de *scaffolding*, aumentando a confiança nos resultados obtidos. Após este processo, os *scaffolds* resultantes podem ser alinhados contra um genoma, permitindo identificar zonas comuns, ou calcular a percentagem de bases alinhadas entre os mesmos. Um dos programas que permite efetuar esta operação é o *Mummer* [90].

2.3.4 Integração de *contigs* ou *scaffolds*

Caso o processamento dos *contigs* e *scaffolds* tenha sido efetuado por programas distintos, ou utilizando diferentes parâmetros, há a possibilidade de integrar os mesmos em *contigs* ou *scaffolds* de maior tamanho. Alguns dos programas desenvolvidos para este fim são o *CISA* [91] e o *GAA* [92].

Por exemplo, no algoritmo implementado são diferenciadas quatro fases: na primeira fase é identificado o *contig* ou *scaffold* de maior tamanho e definido como sendo o *contig* representativo; na segunda fase os *contigs* onde ocorreram erros durante a assemblagem são eliminados ou divididos; na fase seguinte são juntos os *contigs* onde há uma sobreposição nas extremidades e é estimado o tamanho das regiões repetidas; por último há uma fusão do *contigs* onde há sobreposição nas extremidades com tamanhos superiores às regiões repetidas.

2.3.5 Identificação das sequências codificantes

No caso das montagem *de novo*, onde o cálculo de *CDS* apresenta uma utilidade acrescida em relação aos estudos *DNA-Seq com referência*, o ponto de partida são os *contigs* ou *scaffolds* calculados, que já se encontram no formato *FASTA*. Em alguns casos, este processo poderá ter como dados de entrada os cromossomas, sendo expectável dessa forma a obtenção de resultados mais fidedignos, tendo no entanto como contrapartida um processo de assemblagem do genoma mais demorado e com custos associados mais elevados.

Os programas para a previsão de *CDS* podem dividir-se em duas categorias, os que necessitam de sequências de organismos idênticos, para treinarem o modelo a partir do qual serão calculadas as *CDS*, e os que fazem o treino não supervisionado do mesmo modelo. Na primeira categoria de programas os mais referidos são o *GLIMMER* [93] e *Prodigal* [94], para previsão em organismos procariotas, e o *GenScan* [95] e o *GeneID* [96], para eucariotas. Na segunda categoria, destacam-se os programas *GeneMarkS* e *GeneMark-ES* [97]. A principal vantagem dos programas pertencentes a esta categoria é o facto de não ser necessário um conhecimento prévio em relação ao organismo a que pertencem as sequências. Por outro

lado, em alguns casos, podem apresentar valores qualitativos inferiores em termos de sensibilidade e de falsos positivos, quando comparados com os da primeira categoria.

O passo seguinte será fazer uma anotação funcional das *CDS*, utilizando-se para isso programas como o *HMMER* [98] ou o *Blast2GO* [99]. O *HMMER* é um método alternativo ao *BLAST*, no que se refere à pesquisa de homologia entre sequências de proteínas. Relativamente ao *Blast2GO*, pode ser considerado um *framework*, onde são utlizados vários recursos, que permite fazer a anotação funcional de várias sequências, através da atribuição de função molecular, processo biológico e componente celular.

2.4 Aplicações para a análise de dados *RNA-Seq*

Com o aparecimento da sequenciação de próxima geração, novos métodos se tornaram o padrão para fazer a análise da expressão genética, identificados por *RNA-Seq*. Estes têm vindo gradualmente a ser adotados em detrimento dos *DNA microarrays* [100], SAGE [101] e MPSS [102], tornando obsoletos os últimos dois [103].

Do ponto de vista bioinformático, a análise de dados *RNA-Seq*, mais especificamente a comparação e análise da expressão genética, é constituída pelas fases de alinhamento, sumariação, normalização, expressão diferencial e a anotação dos genes de interesse. Tal como os estudos de *DNA-Seq* onde se faz o alinhamento contra uma referência, as leituras são passíveis de serem filtradas através da qualidade das bases (ponto 2.2.1), e a visualização dos dados (ponto 2.2.5) é feita de forma semelhante, sendo no entanto importante ter em conta a anotação da sequência de referência, na altura da análise visual dos dados.

2.4.1 Alinhamento

Para se efetuar o alinhamento dos dados de *RNA-Seq*, podem ser igualmente utilizados os programas referidos no ponto 2.2.2. O processo de conversão das amostras de *RNA* em *cDNA*, de forma a tornar possível a sua sequenciação, juntamente com o facto dos dados se encontrarem também eles no formato *FASTQ*, torna possível a utilização dos mesmos. No entanto, não é recomendada esta abordagem, principalmente leituras originadas por amostras de *RNA* provenientes de organismos eucariotas, dados os eventos de *splicing* alternativo.

As principais necessidades de um *software* de alinhamento para dados *RNA-Seq* são o alinhamento das leituras em zonas onde ocorre *splicing* alternativo, a possibilidade de abertura de falhas, alinhar leituras *PE* e fazer todo este processamento em tempo razoável [104]. Tendo em conta estes critérios, os programas mais indicados para o alinhamento deste tipo de dados são o *TopHat* [105], *TopHat2* [106], *MapSplice* [107], *SpliceMap* [108], *RUM* [104]

e o *GMAP* [48] (Tabela 2-6). O método utilizado para calcular os locais onde ocorre *splicing* é único em cada um dos *softwares*. Do ponto de vista dos utilizadores, com base num questionário presente no *site SEQanswers* [109], o *TopHat* destaca-se claramente como o mais utilizado, com cerca de 70% de utilização, num universo de 80 respostas. Destaca-se ainda o *Bowtie*, com 15%, que provavelmente ainda é utilizado em estudos com organismos procariotas.

O alinhamento no *TopHat*, um dos *softwares* mais utilizados para este processo, ocorre em duas fases. Inicialmente, é feito o alinhamento das leituras no genoma de referência, com o *Bowtie* ou o *Bowtie2*, e são identificadas e indexadas as leituras que não foram alinhadas neste processo. É ainda montado um consenso das regiões com cobertura, inferindo assim putativos exões. De seguida, todos os sítios de *splice* canónicos[13] presentes nos exões vizinhos são enumerados e as sequências derivadas dos possíveis pontos de junção são obtidas. Na segunda fase, as leituras não alinhadas na primeira fase, são alinhadas contra as sequências obtidas no processo anterior. De referir que nas versões mais recentes do *TopHat*, é possível fazer um alinhamento prévio contra um transcriptoma de referência, facilitando todo este processo.

Tabela 2-6 Tabela com características dos programas de alinhamento de dados RNA-Seq. A coluna Plat.Seq refere-se à compatibilidade com os dados das máquinas de sequenciação de próxima geração (I-Illumina, So-ABI Solid, 4-454). As colunas Entrada e Saída são referentes aos ficheiros de entrada e saída. O S, na coluna Falhas, indica que o programa é compatível com a abertura de falhas. A coluna Para.-Paralela indica se o programa suporta computação em paralelo. A QL indica se durante o alinhamento é tida em conta a qualidade das leituras, enquanto a Splicing indica como são detetados os locais de splicing, se de novo ou lib, caso o utilizador tenha que detalhar os locais. Tabela adaptada de [53].

Programa	Plat.Seq.	Entrada	Saída	Falhas	Para.	QL	PE	Splicing	Cit./ano
GSNAP	I,So,4	FASTA/Q	SAM	S	S	N	S	Lib. e de novo	31
MapSlice	I	FASTA/Q	SAM, BED	S	S	N	S	de novo	28
RUM	I,4	FASTA/Q	SAM, TSV, BED	S	N	N	S	de novo	2
SpliceMap	I	FASTA/Q	SAM, BED	S	S	N	S	Lib. e/ou de novo	30
TopHat	I	FASTA/Q, GFF	BAM	N	S	S	S	de novo	121

Uma alternativa aos métodos anteriores é a montagem do transcritoma *de novo*. A principal vantagem desta alternativa é o facto de não estar dependente de um genoma de referência, permitindo dessa forma fazer estudos de expressão genética em organismos onde o genoma não está totalmente construído. Em contrapartida, há a necessidade das bibliote-

[13] Sítios onde o *splicing* geralmente ocorre. Caracterizados pelos dinucleótidos "GT-AG". Existem igualmente *splice sites* não canónicos, geralmente identificados por "GC-AG" e "AT-AC".

cas das leituras terem uma maior cobertura, sendo geralmente o mínimo 30x, e também a necessidade de maior poder computacional para esta estratégia. Num estudo feito por Zhao [110], foram referidos o *Trinity* [111], o *Oases-MK* [112] e o *trans-ABySS* [113], como sendo os que melhores resultados apresentam. O princípio de funcionamento dos mesmos é em tudo semelhante ao da montagem *de novo* de leituras *DNA-Seq* [114].

2.4.2 Sumariação

A partir do ficheiro resultante do alinhamento, o próximo passo é resumir as leituras numa unidade com algum significado biológico, nomeadamente exões, isoformas ou genes. Geralmente, a abordagem mais utilizada é contar o número de leituras que são mapeadas nos vários exões de um gene, mais concretamente a contagem das leituras que estão alinhadas em relação a um dado exão, presente nas várias isoformas (contagem da interceção de exões), ou a contagem das leituras alinhadas aos exões das diferentes isoformas (contagem da união de exões) [115] (Figura 2-9). O objetivo final será a contagem ao nível dos genes, e a soma da expressão ao nível das isoformas, abordagem idêntica à efetuada nos *DNA microarrays* [116]. O programa *Alexa-seq* [117] aplica este método para fazer a sumarização das leituras. Um conceito diferente é aplicado pelo programa *Cufflinks2* [118], já que as leituras são resumidas ao nível das isoformas.

Tendo em conta o questionário do site *SeqAnswers* já mencionado, grande parte das escolhas recai sobre a primeira versão do *Cufflinks*, com cerca de 60%, destacando-se ainda um programa pouco referido nos artigos consultados, chamado *HTSeq-count* [119].

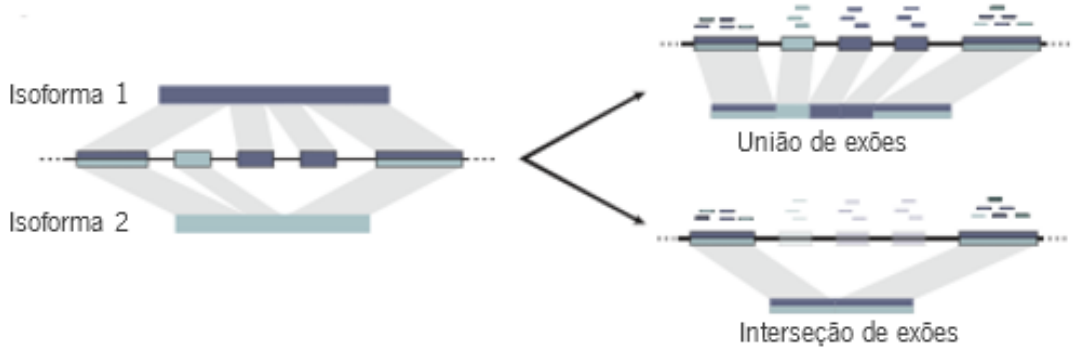


Figura 2-9 Métodos de contagem das leituras. Imagem adaptada de [115].

2.4.3 Normalização

A normalização dos dados é um passo de elevada importância, uma vez que é a única forma de garantir que a comparação dos valores de expressão genética das várias amostras é precisa e válida. A forma de normalização mais simples, e que é igualmente a mais utilizada, é

o ajustamento tendo em conta o número de leituras. A razão pelo qual é necessário este tipo de normalização é o facto da profundidade de sequenciação poder variar entre as várias bibliotecas de leituras, o que iria influenciar o número de leituras alinhadas a um dado exão ou isoforma, não representando no entanto uma expressão mais elevada [34].

Apesar da inúmera quantidade de métodos desenvolvidos para a normalização dos dados, ainda não existe um método unânime aplicado por todos os programas desenvolvidos para este processo. Esta falta de consenso tem sido propícia ao aparecimento de métodos derivados dos já existentes, como de métodos com abordagens diferentes. Os métodos abordados e comparados por Dillies [120] são: o *TC* (*Total count*), *UQ* (*Upper Quartile*), *Med* (*Median*), *DESeq*, *TMM* (*Trimmed Mean of M-values*), *Q* (*Quantile*) e o *RPKM* (*Reads Per Kilobase per Milion mapped reads*). No estudo destes vários métodos, concluiu-se que, por exemplo o ajustamento tendo em conta o número de leituras alinhadas, referido por *TC*, é um método ineficaz. Por outro lado, destacam-se positivamente os métodos *TMM*, aplicado no programa *edgeR* [121] e o *DESeq* [122] aplicado no programa com o mesmo nome.

2.4.4 Expressão Diferencial

O processo pelo qual são selecionados os genes que são diferencialmente expressos, nas diferentes condições, é a análise da expressão diferencial. Métodos anteriormente utilizados na análise de dados de *DNA microarrays*, na teoria poderiam ser diretamente aplicados a estudos de *RNA-Seq*. No entanto, estes não têm em conta vários fatores exclusivos aos dados *RNA-Seq*, como por exemplo, o tamanho da biblioteca das leituras, a cobertura e o tamanho de cada gene [115] [34].

Os primeiros métodos concebidos para a análise de dados de *RNA-Seq* eram baseados na distribuição de Poisson, no entanto, sobretudo pelo facto destes não terem em conta a variação biológica existente entre as amostras, foram substituídos nos últimos anos. O exemplo de um programa que utiliza a distribuição de Poisson é o *DegSeq* [123].

Noutra categoria de programas, encontram-se os baseados na distribuição binomial negativa, que procuram ter em conta a variação biológica entre as amostras. Programas utilizados para fazer a normalização, como o *EdgeR* e o *DESeq* apresentam esta funcionalidade. Existe ainda o *Cuffdiff*, programa pertencente ao conjunto de programas do *Cufflinks*. Tendo em conta o questionário do *site* SeqAnswers, não há unanimidade em relação ao programa a utilizar, dependendo muito do programa utilizado para fazer a sumarização e normalização. O *DESeq*, com 45% das escolhas e o *Cuffdiff*, com 35%, são os que mais se destacam.

É no entanto importante referir que apesar desta ultima categoria de métodos oferecerem alguma significância estatística para o estudo da expressão diferencial entre amostras, é contudo necessário ter alguma ponderação nas conclusões do ponto de vista biológico a que se chegam. É necessário analisar de forma cuidada os resultados, uma vez que a expressão de um dado gene pode variar, por exemplo, pela simples alteração do protocolo na sequenciação das amostras, e também pela variabilidade inerente a todas as amostras biológicas [115]. Uma das formas de atenuar estes efeitos é replicando a sequenciação das amostras, dando origem a várias bibliotecas para uma única amostra.

2.4.5 Anotação Funcional

A anotação funcional, tal como no processo de anotação referido no ponto 2.2.3, pretende atribuir um significado biológico aos genes. No caso dos estudos *RNA-Seq* este processo caracteriza-se pelo agrupamento dos genes que partilham propriedades biológicas entre si, sendo posteriormente feita uma análise de modo a verificar as categorias presentes no conjunto de genes diferencialmente expressos. Pode-se dizer portanto que a principal diferença é o facto de os genes serem comparados em conjunto.

No questionário do *site* SeqAnswers, é exemplificada a possibilidade de utilização de alguns programas que foram desenvolvidos para o estudo de *DNA microarrays*, e que, principalmente pelo conhecimento já adquirido na utilização dos mesmo, continuam a ser usados, apesar da existência de programas com funcionalidades indicadas especificamente para dados *RNA-Seq*. Por exemplo, o programa *DAVID* [124] e o *EasyGo* [125], segundo o questionário, continuam a ser dos mais utilizados, tendo no entanto sido desenvolvidos a pensar em dados de *microarrays*. No entanto, o programa mais utlizado, o *GOseq* [126], trata-se de um *software* desenvolvido tendo em conta os dados de *RNA-Seq*.

2.5 Plataformas integradas para a análise de dados de *NGS*

Um dos serviços mais conhecidos para análise de dados de sequenciação de próxima geração é o *Galaxy* [127]. A principal particularidade desta plataforma é o facto de ser *web*. Esta disponibiliza, entre um conjunto de outras, a maioria das ferramentas apresentadas anteriormente, tanto para análise de *DNA-Seq* com organismo de referência, como para a análise de dados de *RNA-Seq*.

A vantagem deste tipo de abordagem é, pelo menos no caso de utilizadores com conhecimentos mais ligados à Biologia, abstrair o utilizador da instalação e configuração de todos os programas necessários. Oferece ainda a gestão facilitada dos dados, uma vez que

os mesmos ficam guardados por tempo limitado no serviço, e permite a utilização deste tipo de programas a utilizadores sem recursos computacionais apropriados para o mesmo. Também será importante referir é a possibilidade de partilha de *workflows* entre os utilizadores, sendo por isso um bom ponto de partida para utilizadores menos familiarizados com a análise deste tipo de dados.

Por outro lado, quando os dados não se encontram inicialmente disponibilizados no serviço, é necessária a transmissão dos dados através da internet. Devido à elevada quantidade de dados que caracteriza os estudos de *DNA-Seq* e *RNA-Seq*, este facto pode constituir um problema. Existe ainda o dilema da confidencialidade, uma vez que não é garantida a mesma para os dados processados e armazenados no serviço, tal como na altura da transferência dos dados para o *Galaxy*. É igualmente importante referir, pelo menos na utilização gratuita, a baixa velocidade em que os dados são processados.

Noutra categoria, há os pacotes associados às várias linguagens de programação, como o *Biojava* [128], *Bioperl* [129], *R/Bioconductor* [130] e o *Bioruby* [131]. A principal limitação destes pacotes, que em alguns casos servem como uma forma de disponibilizar os programas num ambiente de programação, é a falta de desenvolvimento dos mesmos. Destes destacam-se positivamente os pacotes em *R/Bioconductor*, uma vez que é necessário respeitar um conjunto de parâmetros, como por exemplo a documentação completa de cada pacote, para que estes sejam disponibilizados, e é igualmente a comunidade mais ativa no que diz respeito ao desenvolvimento e atualização dos diversos pacotes. Por outro lado, alguns dos projetos feitos em *Perl* e *Ruby* são abandonados a meio, estão desatualizados ou não documentados, uma vez que são totalmente dependentes da comunidade, e da pessoa que os está a desenvolver. No entanto, pacotes como o *Bioruby* têm definidas estruturas de dados para os principais formatos de ficheiros de sequenciação de próxima geração, sendo por isso de grande utilidade, apesar das limitações.

2.6 Bases de dados

Tem existido um grande esforço em catalogar e disponibilizar de forma livre os dados referentes aos estudos de *DNA-Seq* e *RNA-Seq*. Atualmente, há três grandes bases de dados referentes a estes estudos, o *NCBI GEO* (*Gene Expression Omnibus*) [132], o *ArrayExpress* (*ArrayExpress Archive of Functional Genomics Data*) [133] e o *DDBJ Omics Archive* [134] (Tabela 2-7). Todas permitem a procura direta dos dados pretendidos, podendo por exemplo, os dados serem filtrados por tipo de estudo, ano ou pelo organismo. Existe igualmente a base de dados *SRA* (*Sequence Read Archive*), onde estão disponíveis leituras provenientes de todas

as máquinas de sequenciação de próxima geração, no formato *SRA* [135]. Este formato é uma forma eficiente de guardar as leituras, havendo no entanto a necessidade de descompactar os dados para *FASTQ*, com o programa *SRA Toolkit*.

Tabela 2-7 Tabela com o número de experiências registadas em cada uma das bases de dados. Valores retirados em Fevereiro de 2013 e em Julho de 2013.

	Microarrays	Seq. De próx geração	Total
ArrayExpress	(31605) 34641	(2633) 3511	(34238) 38152
DDBJ Omnis Archive	-	(2098) 2340	(2098) 2340
NCBI GEO	(27641) 29818	(1172) 1550	(28813) 31368

2.7 Exemplos de aplicações

Como referido no capítulo 1, a sequenciação de próxima geração tem as mais variadas aplicações. Neste ponto serão referidos alguns dos artigos alusivos a estas aplicações.

2.7.1 *DNA-Seq*

Relacionados com este tipo de estudos, geralmente os tópicos abordados estão direta ou indiretamente relacionados com a área da saúde. Alguns dos estudos que se destacam, são:

- Cirulli e Goldstein [136], num artigo do ano 2010 apresentam estratégias para a utilização da tecnologia de sequenciação de próxima geração na identificação de polimorfismos de nucleótido único que estão diretamente relacionados com doenças comuns, tendo DePristo *et al.* [137] desenvolvido igualmente uma *framework* para o cálculo de *SNPs* em genomas humanos. Meyerson *et al.* [138] expõe a utilidade desta tecnologia em estudos relacionados com o cancro, e Brase *et al.* [139] faz um estudo idêntico direcionado para a área das doenças neurológicas.
- Relativamente ao uso de *NGS,* no que diz respeito à microbiologia, MacLean *et al.* [140] apresenta um conjunto de casos de estudo relacionados com assemblagem e construção de genomas de organismos nunca sequenciados e o estudo de polimorfismos em estirpes de um dado organismo.
- Relacionado com a sequenciação do genoma humano, Li *et al.* [78] mostra de que forma foi utilizada a sequenciação de próxima geração para fazer a montagem do genoma humano de um africano e de um asiático, de uma forma economicamente viável.
- Mardis *et al.* [141] num artigo de 2011, faz uma retrospetiva do que foram os primeiros anos desta tecnologia, e traça um cenário relativo à mesma.

2.7.2 *RNA-Seq*

O *RNA-Seq* veio igualmente encontrar um mercado e comunidade científica seriamente habituada a uma tecnologia, os *DNA microarrays* [100]. Esta continua a ser até aos dias de hoje a tecnologia mais escolhida quando se pretendem fazer estudos de expressão genética.

Comparando-se as tecnologias poderá afirmar-se que as principais vantagens dos *DNA microarrays* face ao *RNA-Seq*, são o facto de ser uma tecnologia mais barata quando se pretende analisar apenas uma parte específica do genoma, a quantidade de artigos publicados onde a mesma foi utilizada, e ainda o número de ferramentas desenvolvidas e amplamente testadas, para a análise de dados de *DNA microarrays*. Por outro lado, o *RNA-Seq* permite fazer estudos de organismos cujo genoma ainda não se encontra sequenciado, uma vez que não está limitado ao reconhecimento de sequências predefinidas. Permite desta forma fazer o estudo de transcriptomas relativos a novos genes e isoformas. Outra das vantagens do *RNA-Seq* é o facto da expressão genética ser quantificada em valores absolutos [142][143]. Raghavachari *et al.* [144] mostra no entanto a necessidade de estudos combinados, como por exemplo em estudos clínicos, onde os *DNA microarrays* levam vantagem, pelo facto da amostra inicial ser muito pequena. Vários estudos comparativos foram feitos com estas duas tecnologias [145], no entanto, nenhuma conclusão definitiva foi tirada dos mesmos, sendo por isso natural o aparecimento de programas que permitem fazer o estudo dos dados de ambas as tecnologias, como por exemplo o *ExpressionPlot* [146].

Tal como o *DNA-Seq*, o *RNA-Seq* apresenta várias aplicações: estudos da quantificação e comparação da expressão genética entre várias condições, mapeamento do local inicial de transcrição, deteção de fusão de genes, caracterização de *sRNA* e identificação de eventos de *splicing* alternativo [147].

- Gonçalves *et al.* [148] e Trapnell *et al.* [118] apresentam *pipelines* para o estudo de dados *RNA-Seq*, nomeadamente para o estudo da expressão genética e para a identificação de novos genes ou isoformas.
- Num artigo de 2008, Mortazavi *et al.* [149] faz o estudo do transcriptoma dos ratos, permitindo verificar a existência de genes não anotados e a existência de eventos de splicing alternativo.
- Nookaew *et al.* [150] faz um comparativo de várias ferramentas utilizadas para o estudo de dados *RNA-Seq*, utilizando para isso dados relativos à *Saccharomyces cerevisiae*, comparando ainda os resultados das várias ferramentas analisadas com resultados obtidos com a tecnologia de *DNA microarrays*.

- Em relação a casos de estudo relacionados com a saúde, Hackett *et al.* [151] mostra a utilidade destes dados através da caracterização do transcriptoma das células *SAE* (*Small airway epithelium*), com a intuição de arranjar uma conexão entre o facto de se fumar e o cancro no pulmão.
- Do ponto de vista industrial, há a tentativa de integrar dados de *RNA-Seq* com outros dados ómicos, mais concretamente a integração de dados de expressão genética em modelos metabólicos, como forma de melhorar a predição *in silico* de fluxos metabólicos [152][153] [154].

2.8 Sumário e desafios

Apesar das muitas vantagens desta tecnologia de sequenciação em relação ao método de Sanger, esta ostenta novos desafios, principalmente no que se refere à análise dos dados. Esta tecnologia possibilita a sequenciação de milhares de bases de *DNA* em paralelo, originando dessa forma grandes quantidades de dados em bruto, como apresentado na Tabela 2-1. Ainda não existe uma forma eficaz de guardar e catalogar os dados, havendo por isso vários formatos em que estes se encontram guardados, nomeadamente em *FASTQ*, *FASTQ* compactado com o *Gzip*, *BAM* e no formato *SRA*. Uma parte dos estudos já se encontra nas várias bases de dados referidas no ponto 2.6, continuando no entanto, a serem difíceis de encontrar muitos dos dados. A necessidade de pessoas especializadas, particularmente para a análise dos dados e interpretação biológica, constitui igualmente um desafio. Outro fator que impede alguns laboratórios de elegerem estes métodos, é a necessidade de um elevado poder computacional para a análise dos dados, razão pela qual começam a surgir soluções *web*, como a referida no ponto 2.5 [155].

No que se refere ao desenvolvimento de um sistema integrado para o tratamento de dados de sequenciação de próxima geração, o principal desafio estará certamente relacionado com a necessidade de interligar os vários programas. Alguns dos referenciados neste capítulo são resumidos na Tabela 2-8. O facto de haver diversas máquinas de sequenciação, que originam leituras muito específicas, aumenta a complexidade deste desafio. A abordagem a adotar será seguramente a adoção de vários programas, apropriados para a análise das leituras provenientes das diferentes máquinas de sequenciação. Esta abordagem levanta, no entanto, novos desafios, uma vez que se torna necessária a estandardização dos dados. Será portanto necessário encontrar um equilíbrio, no que se refere ao formato dos vários ficheiros e às várias tecnologias de sequenciação, para que exista a possibilidade de analisar os dados de uma forma semiautomática. É igualmente fundamental, que sempre que possível, os parâmetros de cada programa sejam determinados automaticamente.

Tabela 2-8 Lista dos programas indicados para cada uma das fases da análise em estudos de DNA-Seq e RNA-Seq.

Fase da Análise	Função / Método	Programa	Referência
Controlo de qualidade	Relatório de qualidade	*FastQC*	[40]
		NGS QC Toolkit	[41]
	Filtragem e corte	*FASTX-Toolkit*	[42]
		NGS QC Toolkit	[41]
		NGSQC	[36]
Alinhamento	Referência (*DNA-Seq*)	*BWA*	[52]
		Bowtie	[50]
		Bowtie2	[51]
		GMAP	[48]
		SSAHA2	[49]
	De novo (*DNA-Seq*)	*SOAPdenovo*	[78]
		Edena	[74]
		Taipan	[80]
		Velvet	[76]
		ABySS	[77]
	RNA-Seq	*TopHat*	[105]
		TopHat2	[106]
		MapSplice	[107]
		SpliceMap	[108]
		RUM	[104]
	RNA-Seq de novo	*Trinity*	[111]
		Oases-MK	[112]
		trans-ABySS	[113]
Visualização de dados	Visualização de dados	*Tablet*	[66]
		MapView	[65]
		Mauve	[68]
		IGV	[67]
Identificação de características (*DNA-Seq*)	Procariotas	*GLIMMER*	[93]
		GeneMarkS	[97]
		Prodigal	[94]
	Eucariotas	*GenScan*	[95]
		GeneID	[96]
		GeneMark-ES	[97]
Identificação de *SNP* (*DNA-Seq*)		*SAMtools*	[31]
		SOAPsnp	[56]
		Sniper	[57]
		GATK	[55]
Análise de *SNP* (*DNA-Seq*)	Comparação de *SNP*	*VcfTools*	[32]
	Filtragem de *SNP*	*SnpSift*	[60]
	Anotação / efeito do *SNP*	*SnpEff*	[61]

		ANNOVAR	[62]
		Vep	[63]
Sumarização (*RNA-Seq*)	Isoformas	*Cufflinks2*	[118]
	Genes	*Alexa-seq*	[117]
		HTSeq-count	[119]
Normalização (*RNA-Seq*)	DESeq	*DESeq*	[122]
	TMM	*edgeR*	[121]
Expressão diferencial (*RNA-Seq*)		*DESeq*	[122]
		Cuffdiff	[118]
Anotação funcional (*RNA-Seq*)	Ontologia genética	*GOseq*	[126]
		EasyGO	[125]
	Homologia	*Blast2GO*	[99]
		HMMER	[98]
Agrupamento de *contigs* em *scaffolds* (*de novo*)		*Bambus*	[83]
		SOPRA	[84]
		SSPACE	[85]
Eliminação de falhas (*de novo*)		*GapCloser*	[88]
		IMAGE	[89]
		GapFiller	[87]
Comparação de genomas		*Mummer*	[90]
Integração de *contigs* (*de novo*)		*CISA*	[91]
		GAA	[92]

Capítulo 3

Análise de dados *DNA-Seq* – Alinhamento contra genoma de referência

3.1 Desenvolvimento do sistema

Para o desenvolvimento do sistema integrado de tratamento de dados *NGS*, tendo em conta a disponibilidade dos programas para a análise deste tipo de dados, a escolha da plataforma computacional de base teria de recair sempre num sistema operativo baseado em Unix/ Linux. Pela facilidade de utilização, configuração e comunidade existente disponível para a solução de problemas, a escolha recaiu sob o *Ubuntu*, tendo a implementação sido realizada mais concretamente na ultima versão (13.04 64bits). Futuramente, pretende-se disponibilizar o sistema para qualquer sistema operativo Linux, por essa razão são apenas utilizadas o mínimo de ferramentas exclusivas ao *Ubuntu*. Há igualmente a necessidade de instalar as linguagens de programação das quais os programas dependem, nomeadamente *Perl*, *Python*, *Java*, *C* e *R*. Grande parte das mesmas já se encontram instaladas por omissão nas distribuições do *Ubuntu*, simplificando a instalação dos programas.

Do ponto de vista estrutural, o sistema de tratamento de dados *NGS* é composto por três camadas (Figura 3-1), podendo-se considerar a utilização através da linha de comandos ou de chamadas ao sistema dos programas por parte de outras linguagens de programação, como a camada mais baixa do mesmo. Esta camada é construída pelo processo de instalação dos vários programas, que consiste no *download*, compilação, configuração e adição do executável ao sistema operativo através de *links* simbólicos. Uma alternativa válida seria a instalação dos programas através do centro de *software* do Ubuntu, solução rejeitada pelo facto das versões dos programas disponibilizadas já se encontrarem desatualizadas. Outra das razões é o controlo das versões instaladas, possibilitando assim o correto funcionamento do sistema de tratamento de dados *NGS* como um todo.

A segunda camada, desenvolvida em *Ruby*, linguagem de programação flexível e orientada a objetos, permite a uniformização dos programas instalados, nomeadamente na forma como

os mesmos são utilizados e na identificação das variáveis. Um exemplo é o parâmetro correspondente ao número máximo de *threads* que cada programa irá utilizar, identificado distintamente por '-c', '-t' e '-l', por exemplo, e que nesta camada será designado por ':threads', permitindo partilhar o valor do mesmo por todos os programas que possibilitem definir este parâmetro.

A última camada será desenvolvida em *Java*, e será de alto nível. Esta permitirá ao utilizador fazer uma utilização semiautomática de todos os programas implementados no sistema, numa utilização orientada ao objetivo final. Haverá desta forma uma abstração em relação ao programas utilizados, sendo a definição da *pipeline* de processamento definida de forma automática, tendo em conta os parâmetros iniciais.

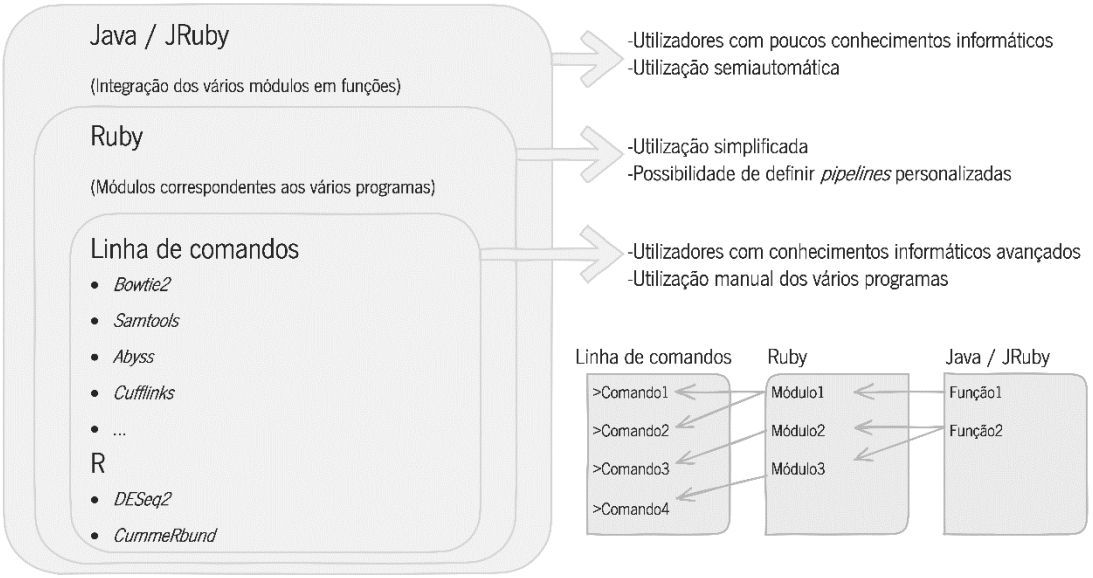


Figura 3-1 Estruturação do sistema em 3 camadas lógicas. Agrupamento dos comandos em módulos, e utilização de vários módulos numa dada função. Permite a utilização simplificada das funcionalidades do sistema, sendo uma função equivalente à utilização de múltiplos comandos na linha de comandos do sistema operativo.

3.1.1 Instalação

No processo de instalação, serão instaladas as últimas versões à data dos programas. É um processo que pode ser feito diretamente pelo utilizador, o *download*, compilação e instalação de cada um dos programas, havendo no entanto a necessidade de respeitar algumas regras em relação à identificação dos *links* simbólicos, de forma a permitir o correto funcionamento do sistema. Por esta razão, e pelo facto da necessidade de se instalar e configurar uma grande quantidade de programas, foi desenvolvido em *Ruby* um *script* que permite a instalação de todos os programas de forma automática. Para o processo de instalação será apenas

necessário definir os locais onde serão guardados os programas, e o local para os *links* simbólicos. Por exemplo, para instalar o *TopHat*, o comando necessário é o seguinte:

```
$ rake install::install_tophat['/usr/local/bin','/usr/local/src']
```

Comando 3-1 Comando para a instalação de um programa específico, neste caso o TopHat. Utilização do install::install_all permite a instalação de todos os programas.

Relativamente ao local a escolher para guardar os programas poderá ser um local à escolha do utilizador. Por outro lado, o local para os *links* simbólicos é aconselhável uma pasta que esteja presente no caminho definido pela variável $PATH do sistema, sendo o '/usr/local/bin' um bom exemplo.

É recomendado, numa fase prévia à instalação, que o utilizador leia os termos e condições de utilização de cada um dos programas, já que os mesmos não são apresentados no momento de instalação.

3.1.2 Módulos e funções

A implementação dos programas na segunda camada foi feita de uma forma modular, com vista à correta organização do código, permitindo a adição de novos módulos, de uma forma relativamente simples. Por outro lado, permite ainda a alteração de módulos já implementados, por exemplo para efeitos de atualização do programa, sem que os outros módulos deixem de funcionar corretamente. Os módulos são implementados em *Ruby*, como referido anteriormente, e são guardados num formato desenvolvido para distribuir pacotes desta linguagem de programação, chamado *gem*[14], neste caso identificado por ngs-0.0.1-java.gem. Esta *gem* pode ser carregada diretamente na *IRB*[15] (*Interactive Ruby Shell*), ou em *scripts Ruby* personalizados. Os módulos por si só não são sinónimo do processamento automático, havendo sempre a necessidade da declaração dos caminhos dos vários ficheiros.

Do ponto de vista da organização dos módulos *Ruby*, cada programa será implementado num módulo identificado pelo nome do programa. Os módulos dos programas podem ser considerados sub-módulos, uma vez que estão inseridos em módulos mais generalistas, nomeadamente o NGS e os DNASeq e RNASeq. Programas utilizados em estudos de *DNA-Seq*, pertencem ao módulo DNASeq, como por exemplo o Bowtie2, enquanto os utilizados em estudos *RNA-Seq* são carregados no módulo RNASeq, como por exemplo o Cufflinks. Por

[14] Pacote que contém os ficheiros necessários para a instalação e utilização das funções implementadas em *Ruby*, e a documentação do mesmo.

[15] Consola interativa que permite definir e chamar funções implementadas em Ruby, sendo os resultados apresentados à medida em que as mesmas são executas.

sua vez, tanto o DNASeq como o RNASeq pertencem ao NGS, tal como o de programas que podem ser utilizados em ambos os estudos, como por exemplo os módulos Utils e Picard.

Por sua vez, os módulos são constituídos por funções que correspondem às funcionalidades disponibilizadas pelos programas. Pretende-se ainda permitir a alteração dos parâmetros correspondentes a cada função, como é exemplo a utilização do módulo Bowtie2:

```
$ bowtie2 –rdg 6 -x reference -1 reads_1.fastq -2 reads_2.fastq -S file.sam
```

Comando 3-2 Comando utilizado para efetuar um alinhamento contra uma referência utilizando-se o programa Bowtie2 na linha de comandos do sistema operativo.

```
>NGS::DNASeq::Bowtie2.align('reference','reads_1.fasta','file.sam',ngs_reads2_path: 'reads_2.fastq',read_gap_open_penalty: 6)
```

Comando 3-3 Funcionalidade equivalente ao Comando 3-2 utilizando-se o módulo implementado.

Como se pode verificar pelos comandos anteriores, sendo o primeiro referente à utilização do programa *Bowtie2* pela linha de comandos e o segundo à utilização dos módulos desenvolvidos, pretende-se que os módulos estejam claramente identificados em relação à sua função, e que os parâmetros apresentem nomes indicativos do que representam. Para utilizadores avançados dos programas, que já têm conhecimento do que significa, por exemplo, o '-rdg', a utilização dos módulos poderá ser relativamente mais lenta, no entanto do ponto de vista do utilizador sem conhecimentos prévios, apresenta muitas vantagens, como por exemplo a fácil leitura do comando, e interpretação das funcionalidades do mesmo. Os parâmetros das funções estão divididos em dois grupos, os obrigatórios e opcionais, havendo a possibilidade, para que seja possível efetuar-se uma utilização semiautomática, destes serem calculados de forma automática. Há ainda a hipótese de atribuir inicialmente valores aos parâmetros opcionais e partilhar os mesmos pelos diversos módulos.

Para o auxílio na utilização dos vários módulos é igualmente disponibilizada documentação de apoio, disponibilizada a partir do *RDoc* da *gem*, que permite verificar as funcionalidades de cada módulo e a descrição de cada parâmetro, como por exemplo a do Bowtie2, acessível com os seguintes comandos:

```
>NGS::DNASeq:: Bowtie2.ri
```

Comando 3-4 Permite a visualização da documentação referente ao módulo Bowtie2, nomeadamente descrição do mesmo e funções implementadas.

```
>NGS::DNASeq::Bowtie2.ri :align
```

Comando 3-5 Documentação relativa à função align do módulo Bowtie2. Permite visualização dos parâmetros de entrada, de um exemplo e de uma descrição geral da função.

Sempre que possível, a documentação é a encontrada nos sítios *web* dos respetivos programas. Tal não invalida a necessidade de uma leitura mais atenta da documentação original, se existir a necessidade de uma melhor compreensão dos métodos e parâmetros a definir de cada programa.

Nos vários módulos referentes aos programas, com vista à automatização e apresentação de resultados na linha de comandos ou no *IRB*, foram desenvolvidas funções e estruturas de dados para ler ficheiros e estruturar o conteúdo dos mesmos em objetos *Ruby*, onde os valores podem ser facilmente acedidos dentro do *Ruby*. Um exemplo, é a fase de controlo de qualidade das leituras, estando dependente a eliminação de leituras repetidas e corte de bases de fraca qualidade das leituras dos resultados estatísticos lidos por uma destas funções.

Como referido na secção 2.5, apesar de não possibilitar o processamento completo dos dados, são de grande utilidade pacotes como o *Bioperl* e o *Biojava*: Neste caso, foi utilizado a *gem bio*, que corresponde ao *Bioruby*, para leitura de ficheiros e estruturação, transformação e exportação dos dados dos mesmos para outro tipo de ficheiros. Um exemplo é o desenvolvimento de funções que a partir dos ficheiros resultantes do *GLIMMER* e do *GeneMark*, e das sequências no formato *FASTA*, constroem ficheiros *GENBANK*. Foram igualmente utilizadas as *gems bio-samtools* e *vcf*, para a leitura de ficheiros *BAM, SAM* e *VCF* e apresentação dos mesmos ao utilizador. Para funcionalidades menos específicas foram igualmente utilizadas as *gems text-table*, que permite a apresentação estruturada de dados em forma de tabelas na consola *IRB*, a *fileutils*, *tempfile* e *ap*, que disponibilizam funcionalidades auxiliares, criação de ficheiros temporários e apresentação simplificada de objetos *Ruby*. Por último, de modo a integrar um interpretador de linguagem *R* no *Ruby*, e correr os comandos *R* no *IRB*, é utilizada a gem *RinRuby*.

É ainda importante referir a utilização de algumas ferramentas do Linux nos módulos, apesar de ser unicamente em casos extremos, onde não foi encontrada nenhuma extensão para *Ruby* com funcionalidades ou desempenho idêntico, como é o caso do *bgzip*, utilizado para arquivar os ficheiros *VCF*, e o *AWK*.

Os módulos DNASeq e RNASeq têm implementadas funções que podem ser consideradas intermédias entre as segunda a terceira camadas, uma vez que disponibilizam funções relativamente globais quanto à sua função, como por exemplo:

```
>NGS::DNASeq.align(sequencia_referencia,leituras,align_program: 'Bowtie2')
```

Comando 3-6 Função genérica que efetua todos os processos necessários para fazer o alinhamento das leituras contra uma referência.

```
>NGS::DNASeq::Bowtie2.build(sequencia_referencia,opções)
>NGS::DNASeq::Bowtie2.align(sequencia_referencia, leituras,sam,opções)
>NGS::DNASeq::SamTools.sam_to_sorted_bam(sequencia_referencia,sam,bam)
>NGS::Picard.alignment_summary_metrics(bam,relatório,opções)
```

Comando 3-7 Conjunto de comandos equivalentes ao Comando 3-6.

A terceira camada apresenta, no entanto, um nível de utilização superior. É implementada em *Java*, e os módulos e funcionalidades desenvolvidos em *Ruby* podem ser utilizados através da plataforma *JRuby*. Para cada um dos tipos de estudos relacionados com *NGS* foi criado um *Env*, conceito desenvolvido pela SilicoLife, e que são representações das análises a efetuar. Estes *Envs*, que em *Java* são representados por classes, pretendem de forma organizada, disponibilizar todas as funções necessárias para um dado estudo, de uma forma simples. Todos os atributos são definidos ao nível da classe tal como os construtores necessários, relativamente aos métodos, sendo desenvolvidos em *Ruby*. Funcionam como uma extensão de uma classe, permitindo desta forma o acesso aos atributos da classe, tanto para leitura como para escrita. Todos os parâmetros, incluindo os não obrigatórios, dos vários módulos implementados em *Ruby* são declarados na classe, permitindo o correto funcionamento das funções. Algumas das vantagens em relação à utilização dos módulos individualmente é o facto da identificação e local dos ficheiros serem gerados de uma forma automática, e ainda a existência de funções para guardar e carregar os *Envs*, permitindo a compactação, transmissão e carregamentos dos ficheiros de uma forma eficaz e independente do utilizador e máquina, desde que o sistema de tratamento de dados esteja corretamente instalado, oferecendo a hipótese de replicar resultados ou de efetuar posteriormente diferentes análises dos dados. Em contrapartida, os *Envs* apenas podem ser carregados numa versão do *IRB* adaptada pela SilicoLife. Um exemplo da utilização simplificada do DnaSeqEnv é a seguinte:

```
>env = SilicoCore::DnaSeqEnv.rnew('nome_projeto','leituras','sequência_referencia',:PROK)
```

Comando 3-8 Inicialização de uma nova instância do tipo DnaSeqEnv. São declaradas os parâmetros obrigatórios, como o nome do projeto, o caminho para as leituras, o caminho para a sequência de referência e o tipo de organismo.

```
>env.set_threads(8)
```

Comando 3-9 Os restantes parâmetros podem ser facilmente definidos com as funções set, havendo ainda a possibilidade de modificar o valor de vários parâmetros ao mesmo tempo com a função load_options().

```
>env.quality_control
>env.align
```

Comando 3-10 O processamento dos dados é feita agora com funções abrangentes que englobam todos os módulos necessários. O utilizador necessita apenas, se achar necessário, definir os parâmetros cada programa, podendo os mesmos ser visualizados com a função print_options().

3.2 Organização do espaço de trabalho

Do ponto de vista de organização do espaço de trabalho nos projetos *DNA-Seq*, há a necessidade de uma estruturação prévia para que o funcionamento do DnaSeqEnv ocorra sem problemas. O nome da pasta principal deverá ser utilizado para identificar o projeto no momento em que este é gerado, sendo recomendado a utilização de um nome que represente o estudo a fazer, por exemplo com a utilização do nome do organismo e data. Aconselha-se igualmente a utilização de uma pasta comum para guardar todos os projetos, sendo possível dessa forma listar os projetos contidos na mesma.

A pasta do projeto será constituída inicialmente por duas pastas (Figura 3-2), nomeadamente a *reads* e a *reference*. Na pasta *reads* são guardadas pelo utilizador as leituras originais, ou em casos em que as leituras são usadas múltiplas vezes, recomenda-se a utilização de *links* simbólicos impedindo dessa forma a redundância dos dados. Durante o processamento do estudo poderão ainda ser criados ficheiros auxiliares, como por exemplo os relatórios de controlo de qualidade, e ainda as leituras processadas pelo programa de controlo de qualidade. A pasta *reference* é o local onde se encontra a sequência de referência, ou sequências, no caso de o organismo ser constituído por mais que um cromossoma. Nestes casos, recomenda-se que previamente se juntem todas as sequências referentes aos cromossomas num ficheiro *FASTA*. Durante o processamento é calculado o índice *BWT* da sequência de referência, originando dessa forma os ficheiros correspondentes ao mesmo.

Na pasta principal do projeto, à medida que os dados são processados, são gerados novos ficheiros, como por exemplo o ficheiro *BAM*, *VCF* e o *VCF* anotado, dependendo das funções utilizadas. Estes ficheiros podem posteriormente ser utilizados por programas que não estão diretamente implementados no sistema de tratamento de dados *NGS*, como por exemplo a visualização dos ficheiros *SAM* e *BAM* no *Tablet*.

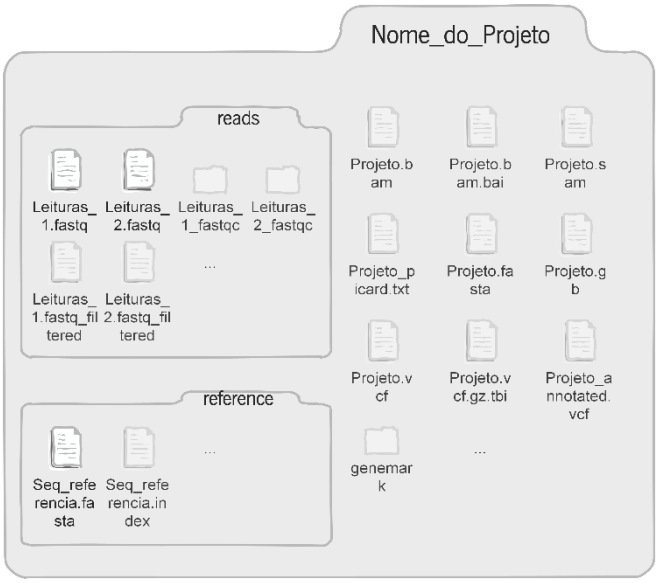


Figura 3-2 Visualização geral do espaço de trabalho de um dado projeto. Os ficheiros inicialmente necessários são as leituras e a sequência de referência, podendo ainda serem necessárias as CDS, no caso de ser necessário utilizar o GLIMMER.

3.3 Organização da *pipeline*

As principais funcionalidades a implementar na *pipeline* para estudos de *DNA-Seq* com referência são: controlo de qualidade das leituras, alinhamento das leituras, identificação de *SNPs*, comparação de *SNPs* e anotação de *SNPs*. Em casos incomuns pode ser também calculada a sequência consenso e calculadas as *CDS* a partir da mesma.

Numa fase prévia ao processamento dos dados, o utilizador necessita sempre de fazer um estudo prévio em relação às leituras que vai alinhar, nomeadamente do organismo a que pertencem. O *download* da sequência de referência é também ela da responsabilidade do utilizador, bem como das *CDS* de referência quando é pretendido usar o *GLIMMER*. Relativamente às leituras a alinhar, pode haver a necessidade de converter as mesmas para o formato *FASTQ*, sendo as leituras originárias da *454* e as da base de dados *SRA* exemplos do mesmo.

3.3.1 Programas selecionados e funcionalidades implementadas

Como fator de seleção dos vários programas foram utilizados critérios como a facilidade de utilização, documentação disponível, tipo de ficheiros de entrada e saída, e consequente facilidade de integração com outros programas, velocidade da análise e resultados obtidos. Estes dados foram adquiridos maioritariamente pela consulta de literatura, nomeadamente os artigos referentes a cada um dos programas, bem como estudos comparativos dos mes-

mos. Os programas que mais se destacaram para cada uma das funcionalidades a implementar foram testados, tendo sido selecionados os seguintes (Figura 3-3):

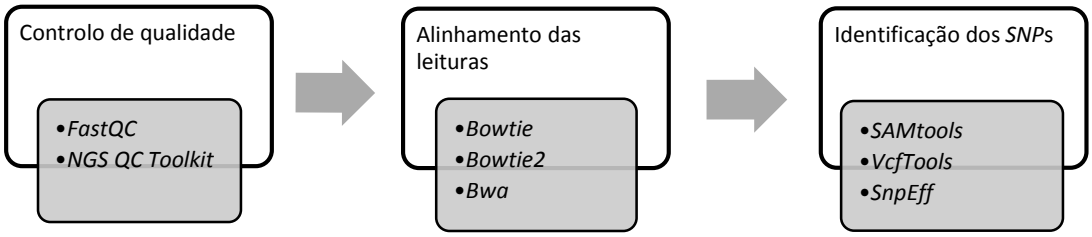


Figura 3-3 Programas selecionados para a implementação da pipeline.

***FastQC*:** programa que permite gerar relatórios de qualidade [*FastQC.report*[16]] das leituras originárias das máquinas *Illumina* e da *454*. Os relatórios são gerados no formato *HTML* (*HyperText Markup Language*) e em formato de texto, permitindo desse modo a leitura e utilização dos valores neles contidos. Algumas das informações de interesse apresentadas no relatório são, por exemplo, o número total de sequência, o tamanho das sequências, a qualidade média das leituras e a percentagem de sequências duplicadas. Foi implementada uma estrutura de dados para representar os relatórios, a {*class FastqcReport*[17]}, que proporciona a leitura dos ficheiros gerados, permitindo a visualização e utilização dos valores de forma individual. A partir da leitura dos ficheiros é igualmente possível gerar um ficheiro com as sequências que apresentam um nível de duplicação elevado [*FastQC. overrepresented_sequences*], permitindo a eliminação das mesmas com o módulo NGS QC Toolkit.

***NGS QC Toolkit*:** permite fazer o controlo de qualidade das leituras provenientes da *Illumina* e *454*, sendo para isso utilizados alguns dos valores calculados anteriormente no módulo FastQC. Este módulo permite ainda definir alguns valores relacionados com o controlo de qualidade, como por exemplo o valor *phred* a partir do qual as leituras são eliminadas ou as bases da extremidade das sequências cortadas, que por defeito é *Q20* [*NGSQCToolkit.illuqc*]. O resultado deste programa é o conjunto de leituras no formato *FASTQ*, filtradas e cortadas nas extremidades [*NGSQCToolkit.trimming*] tendo em conta os valores definidos e um relatório acerca do processamento e da qualidade global das leituras, que é lido e estruturado na {*class NgsqctoolkitReport*}, podendo de seguida ser visualizado.

[16] Módulos e funções implementadas serão identificados por [*módulo.função*]

[17] Nome das classes serão identificadas por {*class nome*}

***Bowtie*:** o *Bowtie* é uma das soluções mais antigas para efetuar o alinhamento de leituras [*Bowtie.align*] de *NGS* contra uma referência. Está implementado sobre ferramentas como o *TopHat* e o *SSPACE,* e suporta o alinhamento de leituras de pequeno tamanho, sendo indicado para quando as leituras têm tamanhos até 50 pb e para análises rápidas. Os indices da sequência de referência poderão ser calculados ou em casos extremos, como o do *H. sapiens* hg18 cujo tamanho é de 2.7 GB, está disponível para *download* no sítio web do *Bowtie*.

***Bowtie2*:** o *Bowtie2* é uma evolução do *Bowtie*, tendo a possibilidade de fazer alinhamentos [*Bowtie2.align*] recorrendo a abertura de falhas. É o programa de alinhamento implementado no *TopHat2*, e mostra ser mais eficaz quando o tamanho das leituras são superiores às alinhadas pelo *Bowtie*. Este programa será utilizado quando as sequências tiverem tamanhos superiores a 50 pb e inferiores a 150 pb. Em contrapartida, este não permite fazer o alinhamento de leituras provenientes das máquinas da *SOLID*, e perde sensibilidade para leituras superiores a estes tamanhos. Tal como no *Bowtie,* estão disponíveis para *download* os índices dos organismos mais utilizados, sendo de notar que as duas versões do *Bowtie* não partilham os mesmos índices, pelo que apenas podem ser utilizados os calculados [*Bowtie2.build*] pela versão do programa correspondente.

***BWA*:** o último programa de alinhamento implementado foi o *BWA*, sendo a versão padrão [*Bwa .aln*] [*Bwa.sampe*] bastante idêntica ao *Bowtie*. No entanto, a sua versão melhorada, o *BWA-MEM* comparativamente ao *Bowtie* e *Bowtie2* mostra ser bastante eficaz no alinhamento de leituras de grandes tamanhos, como as oriundas da *454*. Deste modo, este será o programa de alinhamento utilizado quando as leituras apresentarem valores superiores aos 150 bp. Tal como nos programas anteriormente referidos, também há a necessidade de se calcular os índices [*Bwa.index*] da sequência de referência. Os vários módulos têm igualmente em comum o facto de terem uma função mais genérica que permite efetuar todo o processamento, desde a construção dos índices até ao alinhamento [*Bwa.ngs_reads_to_sam*]. O ficheiro resultante do *BWA*, bem como do *Bowtie* e *Bowtie2*, é um ficheiro *SAM*, obedecendo dessa forma ao formato consenso para alinhamentos.

***SAMtools*:** o *SAMtools* é um programa com elevado número de funcionalidades, sendo por isso implementadas diversas funções no módulo correspondente. Das funcionalidades implementadas destaca-se a função [*SamTools.sam_to_sorted_bam*] que tal como o nome indica, faz o processamento e transformação de um ficheiro *SAM* em um ficheiro *BAM* ordenado e indexado. Esta função refere-se a um conjunto de outras funções, nomeadamente as [*SamTools.faidx*], [*SamTools.import*], [*SamTools.sort*] e [*SamTools.index*], e tem o objetivo de simplificar a utilização e processamento dos dados. Após a conversão para o ficheiro *BAM,* é

possível carregar o ficheiro para uma estrutura de dados [*Utils.get_bam_object*] implementada na *gem bio-samtools*, que permite, entre outras funcionalidades, obter objetos do tipo *Bio::DB::Alignment* de regiões específicas e obter informações em relação à cobertura nas mesmas. Foram ainda implementadas as funcionalidades para fazer a junção de vários ficheiros *BAM* [*SamTools.merge*], o cálculo da sequência consenso do ficheiro *BAM* [*SamTools.sorted_bam_to_fasta*], a cobertura média [*SamTools.avg_coverage*] e máxima [*SamTools.max_ coverage*] em relação ao número de leituras e o cálculo de *SNPs* [*SamTools.sorted_bam_to_ vcf*]. Para os ficheiros *VCF*, com o auxílio da *gem vcf*, que permite representar as linhas dos ficheiros *VCF* correspondentes a um *SNP* ou *SNV* de um modo estruturado, foi também implementada uma estrutura de dados {*class VcfReport*} que representa um ficheiro *VCF*. Esta estruturação permite que facilmente sejam implementadas funcionalidades, como por exemplo a pesquisa de *SNPs* tendo em conta diversos parâmetros.

***VcfTools*:** o programa *VcfTools* permite que de uma forma acessível sejam comparados ficheiros *VCF*. Para esse efeito, foram implementadas funções para comprimir e indexar os ficheiros *VCF*, e para a comparação propriamente dita, nomeadamente a [*VcfTools.compare*] para gerar um relatório relativamente à comparação com os dados necessários para a construção de um diagrama de *Venn*, a [*VcfTools.intersections*] que origina um novo ficheiro *VCF* com os *SNPs* comuns a ambos os ficheiros originais, e a [*VcfTools.complements*] que permite o cálculo dos exclusivos de cada um dos ficheiros. Para o relatório gerado, de modo a que facilmente sejam visualizados os dados contidos no mesmo, foi implementada a {*class VcfCompareReport*}.

***snpEff*:** a anotação dos *SNPs* é feita com o programa *SnpEff*. Este programa apresenta uma base de dados de cerca de 8500 organismos, possibilitando fazer a anotação dos *SNPs* pertencentes aos mesmos. A anotação permite saber, por exemplo, os cromossomas e genes onde ocorrem, e a severidade da alteração provocada. Para este programa, foram criadas funções para fazer o *download* dos dados de um certo organismo [*SnpEff.download*], uma função para listar e pesquisar os organismos disponíveis [*SnpEff.databases*], a [*SnpEff.dump_chromosomes_names*] que permite visualizar a identificação dos cromossomas para que a sequência de referência tenha identificações iguais e o [*SnpEff.annotate*] que permite fazer a anotação dos ficheiros. Os ficheiros resultantes estão no formato *VCF*, pelo que todas as funcionalidades já referidas estão igualmente disponíveis.

***Picard*:** tal como o *SamTools*, este programa tem um elevado número de funcionalidades. Só algumas destas foram implementadas no sistema, destacando-se a [*Picard.alignment_summary_metrics*], que serve para gerar relatórios relativos ao alinhamento,

permitindo obter informações, por exemplo, acerca do número de leituras alinhadas e qualidade do alinhamento, as [*Picard.fix_mate_information*], [*Picard.clean_sam*] e [*Picard.add_or_ replace_read_groups*], que servem para reparar ficheiros *SAM* problemáticos e para adicionar informação aos mesmos e a [*Picard.insert_size_metrics*] para o cálculo da distância entre as leituras *PE*. Tanto o relatório relativo ao alinhamento, como o ficheiro com informações relativas à distância entre as leituras *PE,* são lidos e disponibilizados ao utilizador numa estrutura de dados {*class PicardReport*} {*class InsertSizeMetricsReport*}, esta permite também a utilização dos valores calculados noutras funções implementadas.

Do ponto de vista do *DnaSeqEnv* foram implementadas funções para as principais funcionalidades necessárias para os estudos *DNA-Seq* com referência. São elas o controlo de qualidade [*Env.quality_control*], alinhamento [*Env.align*], cálculo de *SNPs* [*Env.snp*], anotação dos *SNPs* [*Env.vcf_annotation*], comparação de ficheiros *VCF* [*Env.vcf_compare*] e cálculo da sequência consenso e predição de *CDSs* [*Env.consensus_and_cds_prediction*]. Foram ainda implementadas funções que permitem verificar os parâmetros já definidos [*Env.print_options*] [*Env.print_option*] e para guardar [*Env.save*] e carregar [*Env.load*] o *Env*, juntamente com os ficheiros associados e parâmetros associados ao mesmo. Ver Figura A-1, no anexo A, para uma visualização geral da utilização dos vários módulos, ou utilização das várias funções do *Env.*

3.4 Caso de estudo - Caracterização de estirpes de *Mycobacterium tuberculosis*

Neste caso de estudo pretende-se mostrar a utilização da plataforma desenvolvida para a análise de dados de *NGS* quando é pretendido fazer um estudo onde existe uma sequência bem caracterizada que possa ser utilizada como referência no alinhamento das leituras sequenciadas.

Para o efeito, serão analisadas as leituras identificadas por HVNG_1 e IHMT_82_09. Estas são originárias de uma bactéria patogénica identificada como *Mycobacterium tuberculosis*, conhecida por ser causadora da tuberculose. É uma bactéria que se tem vindo a adaptar ao longo do tempo aos fármacos, havendo estirpes multirresistentes e extensivamente resistentes, sendo de grande importância a caracterização das mesmas e a identificação das mutações do genoma que lhe conferem tais características. Estes dados foram gentilmente cedidos pelo Instituto de Higiene e Medicina Tropical em Lisboa.

Nesta análise, serão contempladas as fases de controlo de qualidade das leituras, alinhamento contra um genoma de referência, cálculo da sequência consenso e das sequên-

cias codificantes (*CDS*), avaliação e anotação dos polimorfismos de nucleótido único e *indels*, e comparação dos mesmos.

Estirpes utilizadas como referência**:** para fazer o alinhamento foi selecionado um conjunto de genomas referentes a estirpes da *Mycobacterium tuberculosis* que se encontram melhor caracterizadas (*Tabela B-1*), tendo sido utilizada a estirpe H37Rv para o estudo de polimorfismos.

Análise**:** Inicialmente, foram inicializados os ambientes de trabalho (Envs) para cada uma das bibliotecas em estudo e para cada uma das sequências a utilizar como referência:

```
>opt = { ngs_reads2: 'reads/s_3_2.fastq', threads: 8, qual_cut_off: 20, genome_db: 'Mycobacterium_tuberculosis_H37Rv_uid170532', fasta_cds: 'reference/NC_018143_cds.fasta' }
>h37rv_hvng = DnaSeqEnv.rnew('reads/s_3_1.fastq' ,'reference/NC_018143.fasta' ,TypeOfOrganism::PROK,opt)
```

Comando 3-11 Inicialização do Env,. São definidas algumas variáveis (threads, qual_cut_off) inicialmente, sendo passadas como parâmetro do construtor. É o ponto inicial de um estudo de DNA-Seq com referência.

De seguida, foi efetuado um controlo de qualidade das leituras de forma a eliminar leituras de baixa qualidade. Para isso foi utilizado o valor de 20 para o parâmetro :qual_cut_off, definido nas opções iniciais. Um breve estudo, não apresentado neste texto, permitiu verificar que neste caso o controlo de qualidade teve um efeito positivo, uma vez que permitiu manter a quantidade de bases com cobertura após o alinhamento, ao mesmo tempo que reduziu o número de polimorfismos encontrados, levando à conclusão de que alguns deles seriam originados pela fraca qualidade de algumas das leituras e das bases que foram eliminadas.

```
>h37rv_hvng.quality_control
```

Comando 3-12 Este comando efectua o controlo de qualidade das leituras do Env. Neste caso, será utilizada uma das opções defininidas inicialemente, mais concretamente o qual_cut_off com o valor de 20.

De seguida, efetuou-se o alinhamento das leituras contra o genoma de referência. A seleção dos genomas de referência foi condicionado pelo facto de estarem ou não presentes na base de dados do programa *SnpEff*. Quando não é definido nenhum dos programas de alinhamento é utilizado por defeito o programa *Bowtie2*.

```
>h37rv_hvng.align
```

Comando 3-13 Comando que faz o alinhamento das leituras. Todos os processos necessários são feitos de forma automática, como por exemplo a indexação da sequência de referência e todos os processos necessários até chegar ao ficheiro BAM correspondente ao alinhamento.

Com a função consensus_and_cds_prediction foi calculada a sequência consenso e de seguida as *CDS*. Por omissão, é utilizado o programa *GeneMark*, podendo no entanto, no

caso dos organismos procariotas, ser utilizado igualmente o *Glimmer*, sendo no entanto necessário um ficheiro com *CDS* de um organismo semelhante.

```
>h37rv_hvng.consensus_and_cds_prediction
```

Comando 3-14 *Constrói a sequência consenso, e dependendo do tipo de organismo do ENV, utiliza o GeneMark ou GLIMMER correspondente para efetuar a previsão de CDSs.*

O último passo foi o cálculo dos polimorfismos e *indels* e anotação dos mesmos. Os polimorfismos foram ainda utilizados para fazer uma breve comparação entre as estirpes referentes às duas bibliotecas em estudo, tendo como base o genoma de referência NC_000962 (H37Rv).

```
>h37rv_hvng.snp
>h37rv_hvng.vcf_compare(h37rv_ihmt.get_vcf_file_path())
>h37rv_hvng.vcf_annotation
```

Comando 3-15 Cálculo dos polimorfismos e comparação com os polimorfismo calculados no Env constituído pela mesma sequência de referência mas como as leituras identificadas por IHMT. Por último é feita a anotação dos polimorfismos.

Resultados**:** pela análise dos resultados obtidos no processo do alinhamento em relação aos vários genomas de referência, dados pela Figura 3-4, destaca-se desde logo a elevada percentagem de leituras que foram alinhadas em relação aos mesmos. Os valores, quando previamente efetuado o controlo de qualidade das leituras, situa-se na ordem dos 99%. Por si só, estes valores não têm significado biológico algum, no entanto destaca-se desde logo a estirpe NC_016934 (UT205) como sendo a estirpe com a qual menos leituras são alinhadas em ambos os casos, e as estirpes NC_009565 (F11), no caso das leituras referentes à biblioteca HVNG_1, e as NC_012943, NC_016768 e NC_018078 (KZN) no que diz respeito à IHMT_92_09, em relação à percentagem de leituras alinhadas.

```
>h37rv_hvng.picard_report
```

Comando 3-16 Este comando permite calcular algumas estatisticas referentes ao processo de alinhamento, entre as quais o número e percentagem de leituras alinhadas.

Foram também calculados dados referentes à cobertura das bases, procurando obter informação referente à semelhança entre as várias estirpes, do ponto de vista da sequência. Serão consideradas bases com cobertura todas as bases que não sejam indefinidas, ou seja, iguais a N. De referir que esta informação não tem qualquer validade do ponto de vista de semelhança filogenética.

```
>h37rv_hvng.consensus.pct_bases_with_coverage
```

Comando 3-17 Cálculo da percentagem de bases com cobertura da sequência de referência.

Com base nos resultados obtidos, pôde-se verificar que as bibliotecas em estudo têm em comum o facto da estirpe NC_017026 (RGTB327) ser a que apresenta o resultado mais elevado em relação à percentagem de bases com cobertura, cerca de 98,6% e 98,4%, respetivamente. Estes valores são referentes a 61507 e 72001 bases N.

Do ponto de vista de análise individual das duas bibliotecas, destacam-se mais uma vez as estirpes KZN no caso da IHMT_92_09, e em relação à HVNG_1 a estirpe NC_017524 (CTRI_2) (Figura 3-5).

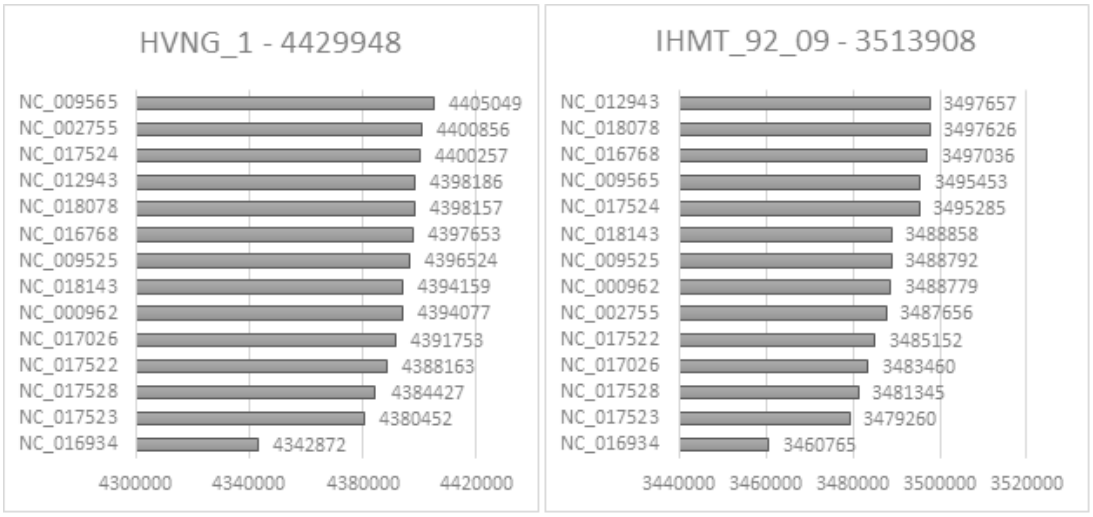


Figura 3-4 Número de leituras alinhadas utilizando as várias estirpes como referência.

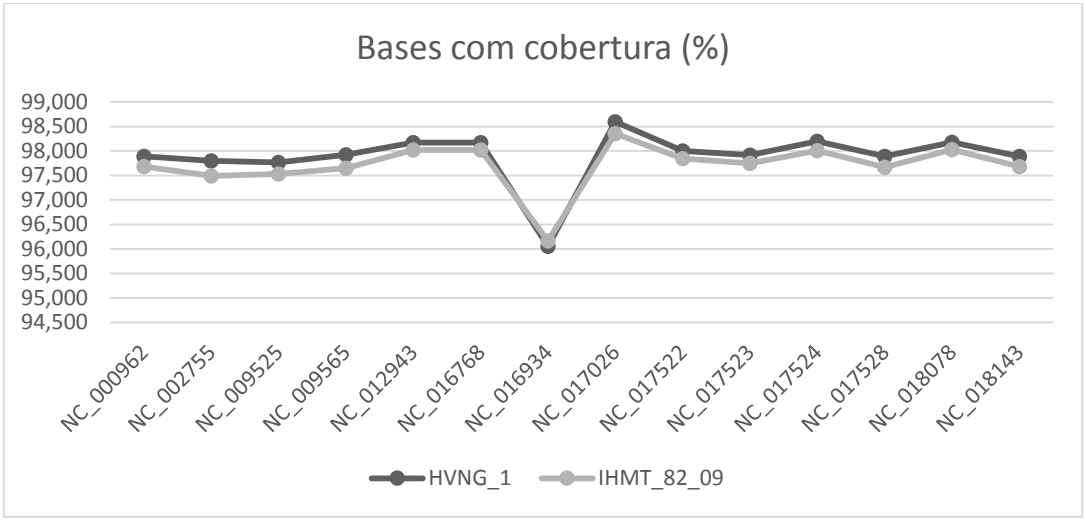


Figura 3-5 Percentagem de bases com cobertura dos diferentes genomas utilizados como referência.

Após o alinhamento das leituras, foi possível obter-se a sequência consenso em relação a cada uma das estirpes usadas como referência. A partir desta, foram calculadas as *CDS*. Analisando os valores de todas as estirpes usadas como referência, foi possível chegar a um valor consenso em relação ao número de *CDS* calculados pelo *GeneMark*. Uma vez que não foi feita uma anotação das mesmas, apenas será referido o número de CDS, que se situa em valores próximos das 4200.

Em relação aos polimorfismos de nucleótido único e indel's, e em concordância com a análise feita na secção referente à cobertura das bases, mais uma vez se pode verificar que os baixos valores de polimorfismos identificados entre as bibliotecas em estudo e as estirpes KZN e CTRI_2 levam à conclusão de que estas estirpes sejam as mais próximas, se tivermos em conta a sua sequência. O número de polimorfismos encontrados são cerca de 600 no caso da HVNG_1, e 700 relativamente à IHMT_92_09. Para efeitos de anotação e de comparação dos polimorfismos entre espécies, como já referido, foi utilizada a estirpe H37Rv, uma vez que é a que se encontra melhor caracterizada e a mais utilizada para efeitos de investigação biomédica. Tendo como ponto de partida os polimorfismos calculados (Figura 3-6), foram calculados os comuns e exclusivos a cada uma das bibliotecas. Conclui-se que estas estirpes têm 696 polimorfismos em comum, dos quais 37 apresentam um impacto elevado no gene onde ocorrem. Estes ocorrem em genes com funções bastante distintas, como por exemplo, funções nos mecanismos de defesa, produção de energia e transcrição.

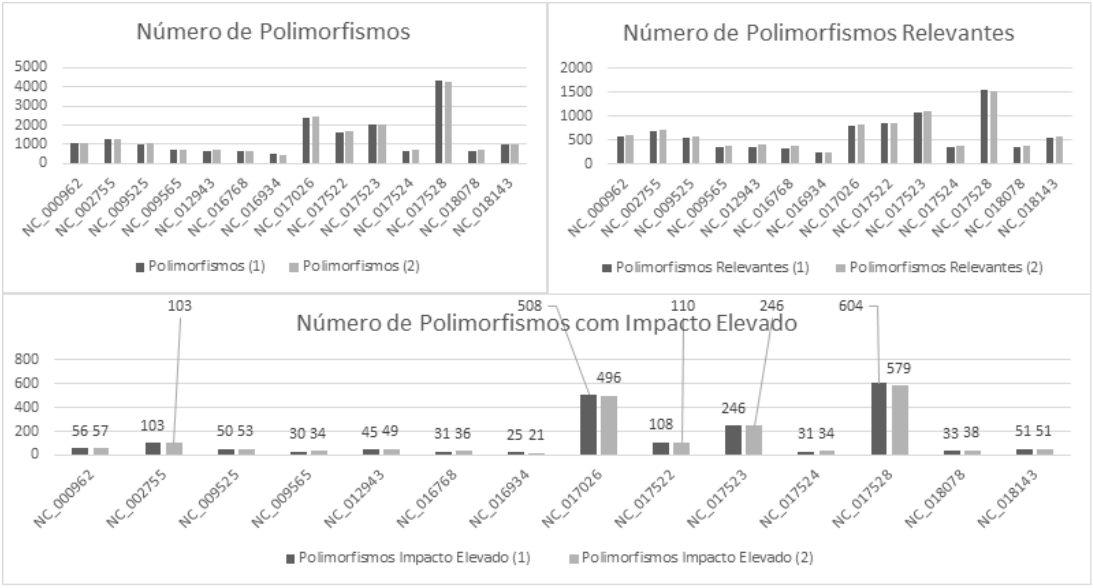


Figura 3-6 Número de Polimorfismos calculados usando as várias referências. ((1) - HVNG_1, (2) - IHMT_82_09)

Para finalizar, foi efetuado um levantamento de genes associados a resistência a fármacos no sítio *web tbdeamdb* [156] e na publicação de Zhang *et al.* [157], sendo posterior-

mente, com a informação obtida, feita uma filtragem dos polimorfismos que ocorrem nesses genes para cada uma das bibliotecas analisadas. Facilmente se encontraram polimorfismos em genes associados a resistência a fármacos, como por exemplo, à isoniazida na biblioteca HVNG_1, à etionamida na IHMT_82_09 e à rifampicina em ambas as bibliotecas.

```
>snp_katG = h37rv_hvng.vcf_snpeff.filter_by_gene('katG')
```

Comando 3-18 Filtragem de uma lista de SNP's referente ao gene katG. Polimorfismos neste gene estão associados à resistência a isoniazida.

Conclusão: Como foi possível verificar, com a utilização de apenas alguns comandos foi possível efetuar o estudo e caracterização de uma forma breve de duas bibliotecas cujos organismos não se encontravam sequenciados e anotados. A principal vantagem desta abordagem é a abstração em relação aos programas utilizados e também a facilidade em ler e perceber a funcionalidade de cada comando.

Do ponto de vista da análise dos dados obtidos, pode-se considerar que das estirpes utilizadas como referência, as mais semelhantes do ponto de vista da sequência são as estirpes *KZN* e *CTRI_2*. Foram também localizados polimorfismos em genes considerados críticos no que diz respeito às características de resistência a fármacos.

Capítulo 4

Análise de dados DNA-Seq – Montagem *de novo*

4.1 Organização do espaço de trabalho

Contrariamente aos estudos DNA-Seq com referência, não há a necessidade prévia de estruturar previamente o espaço de trabalho, havendo unicamente a necessidade de todos os ficheiros relativos às leituras se encontrarem no mesmo local, sendo todos os ficheiros processados guardados igualmente neste espaço. No entanto, tal como no DNA-Seq com referência, recomenda-se a utilização de uma nomenclatura estruturada para que facilmente sejam identificados os estudos correspondentes aos ficheiros.

Os ficheiros inicialmente necessários são as leituras, como já referido, sendo processados um número variável de montagens de genomas a partir das mesmas. Os ficheiros resultantes do processo de montagem dependem do programa utilizado, sendo, no entanto, comum o ficheiro *FASTA* referente aos *contigs* ou *scaffolds.* Estes são utilizados para as restantes análises, como a extensão e eliminação de falhas. O cálculo de *CDS* é processado utilizando o ficheiro referente aos *scaffolds* ou aos *contigs*, originando um ficheiro *GENBANK*.

4.2 Determinação da pipeline

Comparativamente aos estudos de DNA-Seq com referência, esta *pipeline* (Figura 4-1) tem em comum o controlo de qualidade das leituras. As restantes funcionalidades implementadas para estudo DNA-Seq *de novo* foram: a montagem do genoma, a integração de *contigs*, a construção de *scaffolds*, a eliminação de falhas, o cálculo das *CDSs* e a exportação das mesmas para o formato *GENBANK* ou *FASTA*. Estão ainda implementadas ferramentas utilitárias para comparação de genomas e orientação de *contigs* tendo como base uma sequência de referência.

As leituras a utilizar neste Env podem ter o formato *FASTQ* ou o formato *SFF* (*Standard flowgram format*) referente a tecnologia da *454*. As leituras no formato *SFF* são apenas utilizadas no caso de ser selecionado o programa *Celera* para a montagem do genoma, sendo

por isso necessário converter a sequência para o formato *FASTQ* quando for pretendido utilizar um dos outros programas.

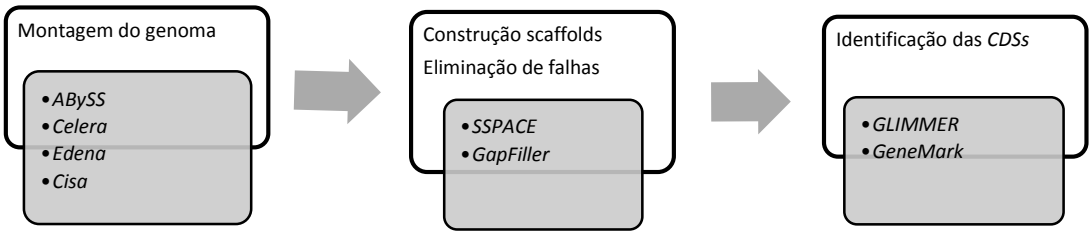


Figura 4-1 Programas implementados para as principais funcionalidades da pipeline.

É igualmente essencial fazer o *download* de uma sequência no formato *FASTA* a usar como referência em casos onde se pretende fazer a ordenação dos *contigs* ou uma comparação dos mesmos com um dado genoma ou conjunto de sequências.

4.2.1 Programas selecionados e funcionalidades implementadas

***GLIMMER*:** o *GLIMMER* é um dos programas mais utilizados quando o objetivo é fazer a previsão de *CDS* em organismos procariotas, sendo por exemplo um dos serviços disponibilizados pela *NCBI (National Center for Biotechnology Information)*. Encontra-se na versão 3, sendo um dos *softwares* mais antigos para esta função. Os resultados indicados no artigo referente ao programa indicam uma eficácia de 98% no que diz respeito encontrar genes, quando comparado com genomas previamente anotados. Estes valores não dizem respeito a encontrar o gene na sua totalidade, e existe ainda um grande número de falsos positivos em algumas situações. Ainda que com estas limitações, não deixa de ser um dos melhores programas para este efeito. Outra das desvantagens é o fato deste necessitar de *CDS* de um organismo idêntico de modo a treinar o sistema, daí haver a necessidade de ter conhecimento prévio em relação ao organismo sequenciado quando este programa for utilizado. Foram implementadas no sistema funções para o cálculo de *CDS* [*Glimmer.glimmer3*] e para, tendo como parâmetros de entrada os ficheiros calculados pelo *GLIMMER* e os *contigs* ou a sequência consenso, criar um ficheiro *GENBANK* com as *CDS* traduzidas [*Glimmer.glimmer_to_genbank*].

***GeneMark*:** o *GeneMark* é um programa com a mesma funcionalidade do *GLIMMER*, tendo como principal vantagem o facto de permitir calcular as *CDS* de organismos eucariontes. É preciso salientar, no entanto, que os níveis de eficácia neste tipo de estimativas é relativamente baixo, sendo portanto necessário um estudo mais aprofundado no que diz respeito ao cálculo de *CDS* e genes hipotéticos em organismos eucariontes. Em relação ao cálculo de *CDSs* em organismos procariotas, este apresenta valores idênticos ao *GLIMMER*, não neces-

sitando no entanto de ficheiros para fazer um treino prévio. Por omissão, este será o programa utilizado para calcular as *CDS*, tendo sido implementadas para este efeito as funções [*GeneMark.ps*] e [*GeneMark.es*]. Tal como no *GLIMMER* é possível extrair as *CDS* e produzir um ficheiro *GENBANK* com as funções [*GeneMark.genemark_ps_to_genbank*] para ficheiros resultantes da análise de apenas um cromossoma, *[GeneMark.genemark_es_to_genbank*] quando o organismo analisado é do tipo eucariota e a [*GeneMark.genemark_s_de_novo_to_genbank*] quando o ficheiro analisado é composto por múltiplas entradas referentes a cada *contig* ou *scaffold*.

ABySS: numa categoria diferente de programas encontra-se o *Abyss*, que é um dos mais utilizados para a assemblagem de dados de *NGS*, juntamente com o *Velvet* e *SOAPdenovo*. Este encontra-se referido em grande parte dos artigos consultados, e em comparação com o *SOAPdenovo* é um programa de fácil utilização. Quando comparado com o *Velvet*, o facto de não ser necessário definir a cobertura esperada e a distância entre as leituras *PE*, fez com que ABYSS fosse implementado em detrimento das duas alternativas. O principal defeito deste é a necessidade de converter as leituras provenientes das máquinas de sequenciação da *454* para o formato *FASTQ*. Por outro lado, permite a utilização de várias bibliotecas ao mesmo tempo, de vários tipos (*PE*, *SE* e *MP*) e tamanhos de leituras. As funcionalidades implementadas foram a montagem de um genoma com apenas uma biblioteca de leituras *SE* [*ABySS.se_assemble*], com vários tipos de leituras [*ABySS.pe_assemble*] e uma função que permite fazer múltiplas assemblagens com diversos valores referentes ao *kmer* [*ABySS.assemble_kmer_optimization*], e que retorna o valor onde foi obtido um valor mais elevado no parâmetro N50. Este valor, um dos valores indicativos da qualidade da montagem do genoma, bem como outros valores, como o número de *contigs* e tamanho do maior *contig*, podem ser obtidos utilizando a função [*ABySS.assembly_stats*], tendo sido criada uma classe {*Class AbyssAssembleStatistics*} que permite aceder facilmente a cada um dos valores calculados. Esta função será utilizada para calcular as estatísticas referentes aos *contigs* ou *scaffolds* sempre que necessário, independentemente do programa utilizado para os gerar.

***Celera*:** o *Celera* utiliza um método diferente para o cálculo dos *contigs*, o *OLC*, referido no ponto 2.3.1. Foi um programa originalmente desenvolvido para utilizar leituras do método de *Sanger*, tendo sido revisto para utilizar leituras da máquina *454* e da *Illumina*. É uma alternativa válida em relação ao *Newbler*, solução distribuída pela 454, tendo como principal falha o facto de só permitir a utilização de leituras com tamanhos superiores a 75 pb, excluindo desta forma uma parte das bibliotecas geradas pelas máquinas da *Illumina*. A principal vantagem é o facto de permitir utilizar ficheiros no formato *SFF*, e utilizar toda a informação contida nos mesmos, por exemplo, para fazer um controlo de qualidade inicial das leituras.

Para fazer a montagem do genoma com este programa foram implementadas as funções [*Celera.fastq_to_ca*] e [*Celera.sff_to_ca*] para gerar os ficheiros com a informação necessária relativa às leituras, e o [*Celera.ca_assemble*] para fazer a montagem propriamente dita. Um dos ficheiros resultantes da montagem é o ficheiro *ASM*, de onde podem ser extraídos os *contigs* calculados [*Celera.asm_output_fasta*].

***Edena*:** este programa tem como base um método para a montagem do genoma idêntico ao *Celera*, sendo derivado igualmente de *OLC*, e pode ser considerado como um complemento do mesmo, uma vez que é indicado para a montagem utilizando leituras de tamanhos pequenos não superiores aos 128 pb. Foi desenvolvido para a assemblagem de pequenos genomas de bactérias, não sendo por isso o mais indicado para genomas de elevada complexidade. A principal razão para a implementação deste método foi a simplicidade de utilização e o facto de serem necessárias pelo menos 3 montagens diferentes de um dado genoma para que seja possível utilizar o programa *Cisa* (ver abaixo). Foram implementadas as funções [*Edena.overlapping*] e [*Edena.assembling*], correspondentes aos dois processos necessários para a montagem do genoma.

***SSPACE*:** o *SSPACE* permite realizar a construção de *scaffolds*, quando disponíveis leituras *PE* ou *MP*. Estas podem ser utilizadas apenas para criar uma ligação entre os vários *contigs* e definir a orientação destes no processo de estruturação dos *scaffolds*, ou podem ser utilizados para a extensão dos *contigs*. Um dos requisitos para a utilização deste programa é a necessidade de saber a distância entre as leituras correspondentes à mesma sequência, valores calculados de forma automática pela função [*Utils.pe_insert_size*]. Para a extensão dos *contigs*, processo útil quando as leituras não são processadas no programa de montagem, é feito internamente com a utilização do *Bowtie*, sendo a construção dos *scaffolds* feito com uma versão melhorada do protocolo implementado no *SSAKE*. O intervalo entre as leituras indicado para o processo de *scaffolding* são dos 200 a 600 pb no caso de leituras *PE* e 2 kb a 10 kb no caso das leituras *MP*. As funções que permitem fazer a construção de *scaffolds* são a [*SSPACE.scaffold*], para a construção do ficheiro de configuração, e a [*SSPACE.scaffold*].

***GapFiller*:** na construção de *scaffolds* são adicionadas *gaps* entre os *contigs*, constituídas por sequências de bases N, com tamanho referente à distância entre os vários *contigs*. Este programa procura preencher as *gaps* originadas neste processo [*GapFiller.close_gaps*]. Este programa deve no entanto ser utilizado com prudência uma vez que pode introduzir artefactos à sequência final. Este processo pode ser usado de uma forma iterativa com o *SSPACE*, permitindo uma diminuição significativa do número de *scaffolds*. Tal como no *SSPACE* é

necessário definir as distâncias entre as leituras para a construção do ficheiro de configuração [*GapFiller.library_file*].

MUMMER: o *MUMMER* permite fazer o alinhamento e comparação entre várias sequências, sendo utilizado para verificar algumas estatísticas relativas à comparação dos *contigs* ou *scaffolds*, relativamente a uma ou várias sequências de referência [*MUMmer.dnadiff*].

Mauve: tal como o *MUMMER*, este programa é um utilitário, não sendo essencial na *pipeline* da análise de dados. Uma das suas funcionalidades permite fazer a ordenação dos *contigs* tendo em conta uma sequência de referência [*Mauve.contig_orderer*].

Cisa: permite efetuar a integração de *contigs* calculados por vários programas de assemblagem de uma forma estruturada [*Cisa.merge_contigs*]. Os valores apresentados no artigo do mesmo mostraram ser bastantes interessantes quando comparados com o *GAA* e restantes programas de montagem de genomas.

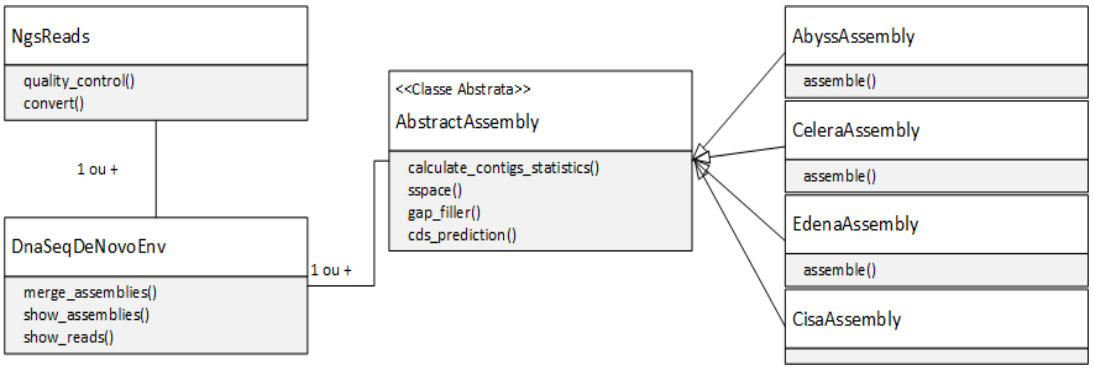


Figura 4-2 Visão geral das classes implementadas essenciais, e funções de cada classe. Como se pode verificar na imagem, um ENV tem que ter obrigatoriamente associado pelo menos um ficheiro com leituras de NGS, e pode ainda ter várias montagens associadas. Cada montagem terá depois associado o ficheiro dos contigs e scaffolds correspondentes ao processamento de dados. As assemblagens adquirem os parâmetros opcionais do ENV, no entanto não partilham opções entre si.

No que diz respeito ao ENV desenvolvido para as análises de DNA-Seq *de novo* é necessário evidenciar algumas diferenças relativamente ao desenvolvido para as análises de DNA-Seq com referência. De uma forma simples, pode-se afirmar que este foi desenvolvido de uma forma orientada à linguagem de programação Java, portanto encontra-se melhor estruturado e adaptado à introdução de novos programas no sistema, sendo um dos objetivos de trabalho futuro a reformulação do ENV de DNA-Seq com referência.

A implementação de classes abstratas (Figura 4-2) para representar alguns ficheiros ou fases da análise, permitiu definir de forma estruturada os ficheiros de entrada e saída de cada fase e as funcionalidades disponibilizadas para cada programa. Um exemplo do mencionado é a classe abstrata AbstractAssembly onde são definidos os atributos comuns a todos

as assemblagens e todas as funções a utilizar nessa fase. Para cada programa de assemblagem foi desenvolvida uma nova classe que adquire todas as características da classe abstrata, como é exemplo a classe *AbyssAssembly*. Algumas das funcionalidades implementadas na classe abstrata são as funções para fazerem a assemblagem, para calcular estatísticas dos *contigs* ou *scaffolds* e para calcular as *CDSs*. As várias assemblagens podem ser adicionadas ao ENV principal, onde estão implementadas funções com vista à utilização das mesmas.

4.3 Caso de estudo – *Pseudomonas* sp. M1

A pseudomonas é uma bactéria aeróbica gram-negativa pertencente à família das *Pseudomonadaceae*. As bactérias pertencentes a este género apresentam uma grande diversidade metabólica, e algumas das quais têm importante utilidade industrial, como é o exemplo de algumas estirpes da *P. fluorescens*, utilizadas no controlo biológico de patogénicos e de estirpes da *P. alcaligenes* e *P. putida*, utilizadas na biorremediação devido a características que lhe permitem metabolizar compostos químicos poluentes.

A *Pseudomonas* sp. M1, em estudos publicados, apresentou algumas características interessantes do ponto de vista industrial, como a capacidade de utilizar como fonte de energia compostos como o fenol e o benzeno. De forma a compreender e caracterizar esta estirpe, esta foi sequenciada e anotada por Soares-Castro *et al.* [158].

Análise: Neste caso de estudo, será utilizado o sistema ao nível das funções disponibilizadas pelo ENV, nomeadamente do ENV referente ao DNA-seq *de novo*. De seguida, os resultados são comparados com os resultados apresentados no artigo mencionado.

As leituras utilizadas para a montagem do genoma, e que são as mesmas utilizadas no artigo, foram gentilmente cedidas pelo professor Pedro Santos (Universidade do Minho), um dos autores do mesmo. Foram disponibilizadas duas bibliotecas, a referente às leituras da máquina *454 FLX*, composta por 264,177 leituras *SE* de tamanho médio de 525 pb, e a biblioteca *PE* originária da máquina *Genome Analyzer IIx*, composta por 5,303,579 leituras de tamanho 2 x 50 pb, com uma distância média entre os pares de leituras de 320 pb. No estudo, será igualmente referido o efeito da não utilização das leituras *MP* referidas no artigo, tanto no número de *contigs* e *scaffolds* a que se consegue chegar, como à anotação dos *scaffolds* em relação à anotação presente no *NCBI*.

O genoma presente no *NCBI* é composto por um único *scaffold*, composto por 9 *contigs*, obtidos utilizando os programas mencionados no artigo e nos ficheiros *GENBANK* (*Newbler*, *MIRA*, *SSPACE*, *GapFiller*, *GapCloser* e *Anchor*). É, no entanto, necessário mencio-

nar dois factos importantes a ter em conta na comparação dos resultados obtidos pelo sistema desenvolvido com os mencionados no artigo, que são a utilização das leituras *MP* para a criação dos *scaffolds*, a curação manual e a utilização de sequenciação de Sanger para a validação dos dados obtidos. O sistema desenvolvido realizará o processo mais simples da montagem e validação de um genoma, que são a assemblagem das leituras num *draft* do genoma. O processo de validação, análise e finalização será sempre o processo mais demorado neste tipo de análises.

Os resultados serão referentes a 3 projetos distintos, correspondentes a um único ENV. No primeiro projeto foi utilizado o *Celera* para fazer a assemblagem das leituras da *454*, tendo sido de seguida utilizados os *contigs* gerados como ponto de entrada do *SSPACE*. As leituras *PE* foram utilizadas inicialmente para estender os *contigs*, sendo, de seguida, utilizadas para a construção dos *scaffolds*. O próximo passo foi, com o *GapFiller*, fechar as *gaps* existentes entre os *contigs*, resultantes da utilização do programa *SSPACE* na construção dos *scaffolds*.

```
>m1_env = SilicoCore::DnaSeqDeNovoEnv.rnew('PseudomonasM1',SilicoCore::TypeOfOrganism::PROK)
>m1_env.addNgsReads(NgsReads.new("m1PE","/ pe-M1_1.fastq","/ pe-M1_2.fastq", NgsReadsType::PE, NgsReadsTechnology::ILLUMINA))
>m1_env.set_threads(8)
>celera_assembly = SilicoCore::CeleraAssembly.rnew("celera_assembly",m1_env)
>opt = {
trim_soft: true,
utg_error_rate: 0.03,
utg_error_limit: 2.5,
ovl_error_rate: 0.06,
cns_error_rate: 0.10,
cgw_error_rate: 0.10,
unitigger: 'bog'
}
>celera_assembly.load_options(opt)
```

Comando 4-1 Inicialização de um ambiente de trabalho, adição das leituras PE e definição do parâmetro correspondente às threads, que será partilhado por todos os projetos. De seguida, é inicializado um projeto do Celera, e são definidos um conjunto de parâmetros que apenas serão utilizados no mesmo.

No segundo, a montagem é feita utilizando-se o *ABySS*, sendo utilizadas tanto as leituras *SE* como *PE* no processo. As leituras *SE* são inicialmente convertidas para o formato *FASTQ*, utilizando-se o programa *Flower* [159], que ainda não se encontra implementado no sistema. O processamento seguinte foi igual ao executado no primeiro projeto.

```
>abyss_assembly = AbyssAssembly.rnew("abyss_41",41,m1_env)
>abyss_assembly.assemble
```

Comando 4-2 Inicialização de um assembly referente ao programa ABySS, onde é passado como parâmetro a identificação e o valor do kmer a utilizar. De seguida, é feita a assemblagem com um único comando.

Por último, foi utilizado o programa *Edena* para fazer a montagem das leituras *PE*. Os *contigs* calculados foram, de seguida, integrados com os calculados pelo *Celera* e *ABySS*. A análise posteriormente feita foi igual à dos projetos anteriores.

```
>m1_env.addAssembly(celera_assembly)
>m1_env.addAssembly(abyss_assembly)
>m1_env.addAssembly(edena_assembly)
>m1_env.merge_assemblies('cisa_assembly')
```

Comando 4-3 As assemblagens calculadas anteriormente são adicionadas ao Env, sendo de seguida utilizado o programa Cisa para fazer uma integração das mesmas.

Resultados: após o cálculo dos *contigs*, as estatísticas relativas à qualidade da assemblagem do *contigs* gerados foram calculadas.

```
>m1_env.assemblies['abyss_41'].contigs_stats
```

Comando 4-4 Depois de adicionados ao Env os projetos podem ser acedidos pela sua identificação. Neste comando são calculadas estatísticas referentes à assemblagem com o programa ABySS.

Relativamente ao número de *scaffolds* (Tabela 4-1), pode-se verificar a capacidade do *Cisa* na integração destes. O *Cisa* apresenta igualmente, tirando o número de bases indefinidas, os valores mais positivos. Após uma breve análise com o comando dna_diff, verificou-se a existência de muitas regiões repetidas nos *scaffolds* calculados, existindo um alinhamento de 87,73% das bases da referência (*contigs* retirados do NCBI). Uma das razões poderá ser a elevada complexidade do genoma da *Pseudomonas* sp. M1 que levou a um comportamento errático do algoritmo do *Cisa*. O *Celera* apresenta valores relativamente melhores que o *ABySS*, apesar de utilizar unicamente as leituras provenientes da *454* na altura de assemblagem. Uma das razões poderá ser o facto da incorporação das leituras *PE* no processamento efetuado pelo *SSPACE*. Uma comparação dos *scaffolds* com os *contigs* do NCBI permitiu verificar uma concordância muito superior à encontrada no *Cisa*, com cerca de 99,65 % das bases alinhadas. De qualquer forma, existem ainda assim aproximadamente 25000 bases não alinhadas, que podem ser o reflexo da não utilização das leituras *MP* e da ausência da curação manual dos *scaffolds*. Os valores relativos ao alinhamento dos *contigs* do NCBI, tendo como referência os *scaffolds* do *Celera*, onde cerca de 1000 bases não obtive-

ram qualquer alinhamento podem significar um conjunto de bases incorretamente alinhadas na altura da montagem.

```
> m1_env.assemblies['abyss_41'].dna_diff('/pseudomonas_m1.fasta')
> m1_env.assemblies['celera_assembly'].cds_prediction
```

Comando 4-5 No primeiro comando, os scaffolds calculados pelo programa Cisa são alinhados e comparados com os contigs retirados do sítio web NCBI. No segundo, é mostrado como é feito o cálculo das sequencias codificantes, utilizando por defeito o programa GeneMark adaptado ao tipo de organismo definido na altura da criação do ENV.

Tabela 4-1 Tabela com estatísticas referentes às várias assemblagens. Destacam-se claramente os valores do Cisa, mas como referido estes valores não são significativos de uma melhor assemblagem. Na comparação entre o ABySS e o Celera destaca-se o facto do Celera ter melhores valores globais, como o N50 e o número de scaffolds.

	Nº *Scaffolds*	N50	n:N50	min	max	soma	Bases N
ABySS	226	55966	36	670	317711	6931046	5153
Celera	132	79140	26	1040	295352	6920396	681
Cisa	73	116539	19	14061	634544	7108920	2798

A etapa seguinte foi a anotação, tendo sido utilizado o programa *Prokka* [160] para este efeito. Este foi utilizado, apesar de não implementado no sistema desenvolvido, de modo a permitir uma correta comparação dos valores, uma vez que o genoma disponível no NCBI foi também ele anotado com este programa. O *Prokka* faz uso do *Prodigal* para o cálculo de *CDS* e de uma série de programas auxiliares mencionados no sítio *web* do mesmo, para a atribuição de uma identificação às várias *CDS*.

Serão apenas comparados os valores relativos à quantidade de cada característica da anotação, como o número de *CDS*, características (*features*), *RNA*, proteínas hipotéticas e números *EC* (*Enzyme Commission*). De uma forma geral, há uma grande concordância nos valores (Tabela 4-2) das anotações dos *scaffolds* gerados, com a anotação presente no NCBI. Estes valores carecem de validação, como por exemplo a verificação se os genes identificados e os números *EC* são iguais.

Tabela 4-2 Valores referentes à anotação dos scaffolds calculados. Há uma conformidade geral entre os valores do ABySS e do Celera. Se compararmos com os valores do ficheiro retirado do NCBI (GenBank) verifica-se uma maior quantidade de CDS calculadas por parte do Celera e do ABySS, podendo o facto do genoma estar mais fragmentado ser uma das razões. Em relação aos números EC há igualmente uma concordância entre os valores.

	CDS	Características	RNAs	Proteínas hipotéticas	Números EC
ABySS	6110	6221	85	1224	1992
Celera	6121	6232	82	1200	2005
Rast	6191	6255	-	1342	1471

GenBank	6053	6136	-	810	-
GenBank (*Prokka* 1.7)	6091	6201	77	1166	2007

Procede-se criação de mapas KEGG (*Kyoto Encyclopedia of Genes and Genomes*), utilizando para isso as sequências das proteínas calculadas pelo *Prokka* como dados de entrada no serviço *web* para anotação de proteínas com os números *EC* (*KEGG Automatic Annotation Server* [161]). Os números observados são de seguida mapeados nos mapas, permitindo observar as reações possíveis num dado mapa. A visualização dos mesmos permite, de certa forma, dar um maior grau de confiança em relação aos *scaffolds* gerados e à capacidade dos mesmos conterem a informação essencial para a criação de modelos à escala genómica dos organismos sequenciados. Como se pode verificar pela comparação dos mapas gerados (Tabela C-2), há uma grande concordância entre os mesmos (*Figura 4-3*). Por outro lado, o número de termos *KO* (*kegg orthology*) associados às sequências é também ele concordante, 3283 no caso das proteínas dos *contigs* do NCBI e 3306 no que diz respeito aos *scaffolds* do *Celera*.

Conclusão**:** De uma forma simples foram feitas quatro assemblagens distintas de um conjunto de dados. A partilha de informação, como por exemplo as leituras, e de parâmetros do Env entre as várias assemblagens permite uma análise dos dados mais simples e rápida no que diz respeito á necessidade de *input* por parte do utilizador. Por outro lado, parâmetros exclusivos a cada tipo de classe dão a flexibilidade de parametrização necessária a este tipo de análises.

Através do cálculo de dados estatísticos referentes aos *contigs* e *scaffolds* procurou-se ordenar as assemblagens em relação à sua qualidade. O facto deste estudo ser com um genoma já assemblado, cujas leituras são as mesmas, permitiu fazer uma comparação dos *scaffolds* gerados, com os *contigs* disponíveis publicamente e já validados, permitindo identificar erros naquela que seria em termos estatísticos a melhor assemblagem, a do programa *Cisa*. Este facto veio reforçar a necessidade de um sentido crítico na análise dos dados e de uma validação dos mesmos.

A anotação do *draft* do genoma permitiu ainda comparar as várias assemblagens em relação à quantidade de características presentes no genoma.

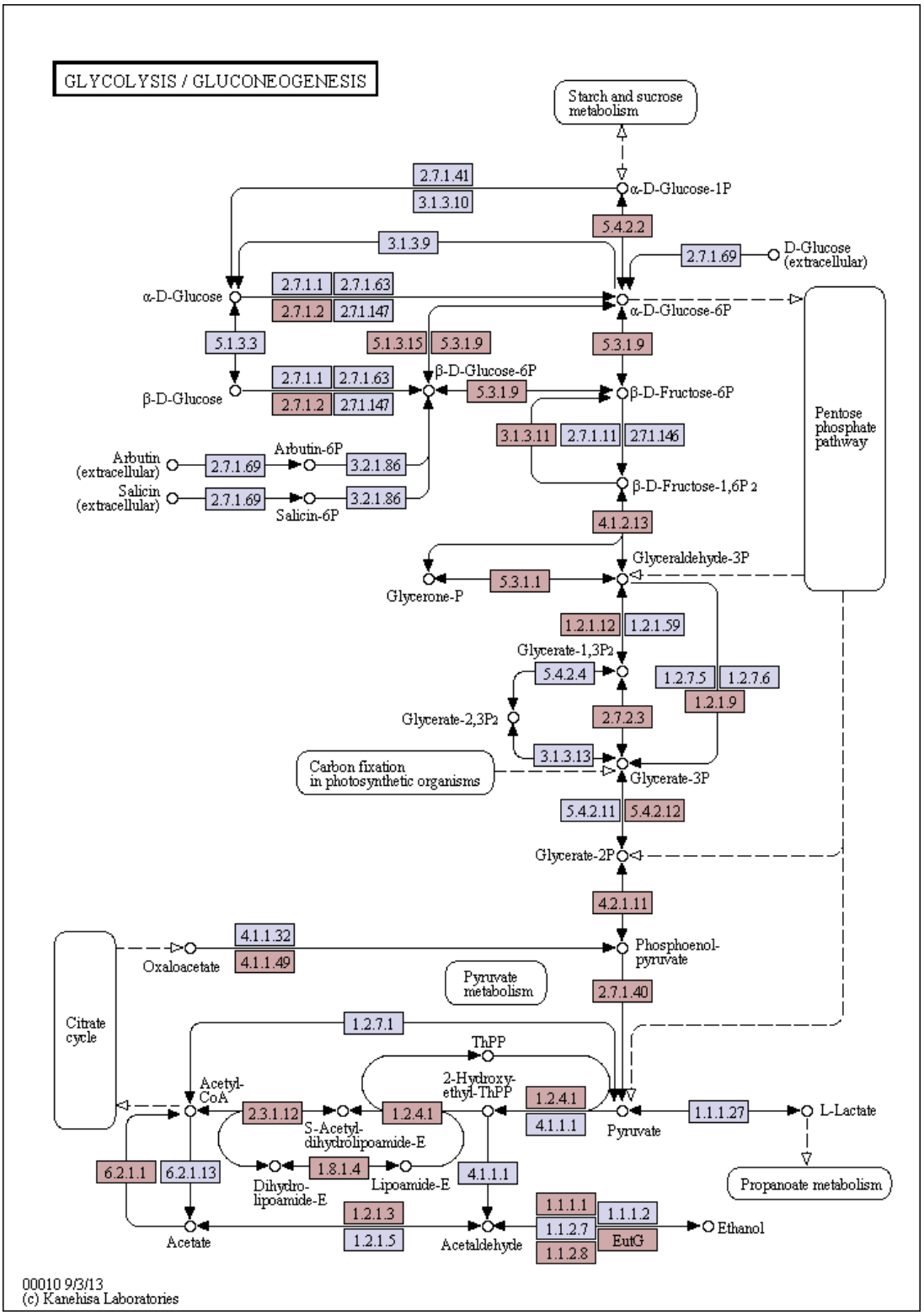


Figura 4-3 Mapa relativo à Glicólise onde é possível ver a existência de poucos gaps. O mapa gerado é igual para o caso da anotação com os dados relativos aos scaffolds gerados com o Celera e dos contigs disponibilizados no NCBI. Encontram-se destacadas a vermelho as proteínas anotadas com um número EC presente no mapa.

Capítulo 5

Análise de dados RNA-Seq

5.1 Organização do espaço de trabalho

Em conformidade com as análises de dados de DNA-Seq, os ficheiros necessários para um dado estudo deverão estar localizados num dado local comum a todos, permitindo desta forma que estes sejam guardados e carregados em outros computadores. Tal como nos projetos de DNA-Seq com referência há a necessidade de sempre que se inicia um novo projeto, da criação de uma pasta principal, e de pastas para as leituras e para as referências.

Neste caso, existirão obrigatoriamente pelo menos quatro ficheiros do tipo *FASTQ* ou *SFF* referentes a duas condições distintas constituídos por duas réplicas referentes a uma dada condição. Estas são referentes a duas recolhas de amostra da mesma condição que são sequenciadas. É ainda necessário definir o genoma a utilizar como referência e a anotação do mesmo no formato *GFF* (*Generic Feature Format*). Quando pretendido efetuar uma anotação dos genes expressos apenas é necessário o ficheiro relativo ao genoma de referência. Todos os dados calculados, geralmente guardados no formato *CSV* (*comma-separated values*), serão guardados na pasta principal.

5.2 Determinação da *pipeline*

O controlo de qualidade, comum aos estudos de DNA-Seq com referência e DNA-Seq *de novo*, também é um dos passos necessários nos estudos deste tipo. As funcionalidades implementadas em exclusivo no RNA-Seq são o alinhamento das leituras contra um genoma de referência e contra a anotação desse genoma, eliminação de moléculas repetidas e estudos da expressão genética. É ainda possível com os programas integrados no sistema fazer um estudo com o objetivo de encontrar novos genes em relação a uma anotação de referência, ou a partir do alinhamento das leituras a um genoma, calcular os transcritos e fazer a anotação dos mesmos.

Numa fase anterior à *pipeline* e à utilização do sistema, há a necessidade de uma fase manual. Nesta, o utilizador terá de fazer o *download* dos ficheiros referentes ao genoma de referência no formato *FASTA* e o relativo à anotação do mesmo, no formato *GFF*. De seguida, sendo este um passo crítico, é essencial que a identificação dos cromossomas seja concordante entre o ficheiro do genoma e o ficheiro da sua anotação, caso isso não aconteça o utilizador terá de manualmente alterar os mesmos para que isso suceda.

Figura 5-1 Programas implementados para as principais funcionalidades da pipeline de RNA-Seq.

5.2.1 Programas selecionados e funcionalidades implementadas

TopHat2: o *TopHat2* é um programa de alinhamento feito exclusivamente para a análise de dados de RNA-Seq. É uma das ferramentas constituintes dum conjunto de ferramentas identificadas como "*Tuxedo Suite*", juntamente com o *Bowtie* e o *Cufflinks*. É considerada pela comunidade de utilizadores uma ferramenta que alia os bons resultados no que diz respeito ao alinhamento, com a velocidade de processamento, e é também um dos programas que melhor documentado se encontra. Outra das razões para a integração deste programa no sistema são a facilidade e a flexibilidade de uso que o mesmo disponibiliza, permitindo fazer um alinhamento de leituras referentes tanto de organismos procariotas como de eucariotas, utilizando-se parâmetros específicos para cada um. Uma alternativa para organismos mais simples é a utilização do *Bowtie2* ou do *BWA*. A principal vantagem do TopHat em relação a estes é permitir o cálculo de zonas de *splicing* alternativo mesmo sem uma anotação de referência. A função utilizada para o alinhamento é a [*TopHat.align*], sendo o ficheiro resultante deste processo um ficheiro do tipo *SAM* ou *BAM*.

HTSeq: o *HTSeq* permite mapear as leituras que foram alinhadas em relação aos componentes constituintes de um genoma anotado, como por exemplo genes, exões ou qualquer tipo de *RNA*. O ficheiro de entrada é um ficheiro *SAM*, havendo a necessidade de converter o ficheiro *BAM* resultante do alinhamento com o *TopHat2*, sendo uma das hipóteses a utilização da função [*SamTools.bam_to_sam*]. É necessário ainda o ficheiro *GFF* utilizado no pro-

cesso de alinhamento, estando o mapeamento dependente da identificação de cada *feature* da anotação. Este programa foi desenvolvido pela mesma equipa que desenvolveu o *DESeq2*, sendo por isso uma escolha óbvia quando pretendido utilizar o *DESeq2* para estudos de expressão diferencial. A função implementada para o processamento com o *HTSeq* é a [*HTSeq.count*], e permite a definição de todos os parâmetros, como por exemplo a escolha do algoritmo para fazer a contagem das leituras.

***DESeq2*:** juntamente com o *Cufflinks* é um dos programas mais utilizados para o estudo de expressão genética. É um dos programas que melhor desempenho mostrou desde as suas versões inicias, tendo mesmo vindo a ser implementados algoritmos desenvolvidos para o mesmo em outros programas para o mesmo efeito. Trata-se de um programa desenvolvido em R pelo que uma das possibilidades é também a sua utilização no *RStudio*, e apresenta funcionalidades de criação de imagens e tabelas com conteúdo de interesse. Com a utilização do *RinRuby* foi possível disponibilizar as funcionalidades no sistema. As funções implementadas foram a [*DESeq2.differential_expression*] para a análise da expressão diferencial. Todas as outras funções estão dependentes desta, já que utilizam os dados calculados pela mesma. Os dados são exportados para um ficheiro no formato *CSV*, ou podem ser explorados por funções auxiliares, como por exemplo, a [*DESeq2.heatmap*] que permitem a criação de imagem com o *heatmap* de um conjunto de genes. Pode também ser importante fazer a normalização dos dados obtidos, por exemplo antes de fazer uma análise de componente principais [*DESeq2.pca*]. Foram integradas as funções [*DESeq2.rlog_transforma tion*] e [*DESeq2.variance_stabilizing_transformation*] para o efeito. É ainda possível filtrar um conjunto de genes tendo em conta os valores correspondentes ao "*fold change*", permitindo identificar os genes cuja expressão sofreu maiores alterações tendo em conta a condição [*DESeq2.de_subset*], ou ainda verificar os genes que não eram expressos numa dada condição e passaram a ser, ou o contrario, com a função [*DESeq2.silenced_or_expressed_subset*], quando são apenas estudadas duas condições.

Cufflinks: programa desenvolvido pela equipa que desenvolveu o *TopHat*. Ao contrário do *DESeq2*, que faz o cálculo da expressão ao nível dos genes ou exões, o *Cufflinks* faz a contagem ao nível dos transcritos que codificam um gene ou as isoformas do mesmo. Este faz uso total dos dados calculados pelo *TopHat*, utilizando os dados referentes ao splicing alternativo processados pelo mesmo, permitindo desta forma o encontrar novos genes ou transcritos, sendo essa a sua principal vantagem em relação aos restantes programas. A primeira versão do programa, apesar de muito utilizada pela comunidade, e de ter sido utilizada vários para estudos de expressão diferencial publicados, mostrava alguns problemas, tendo um elevado

número de falsos positivos e falsos negativos. No entanto, na versão atual estes foram corrigidos através da adoção de métodos desenvolvidos para o *DESeq2*. As principais funcionalidades são a identificação de locais referentes aos genes e locais de splicing alternativo gerando um ficheiro *GFF* [*Cufflinks.cufflinks*], bem como os valores correspondentes ao *FPKM* (*fragments per kilobase of exon per million fragments mapped*) ao nível dos genes e isoformas. Permite ainda a integração de vários ficheiros *GFF* calculados pela anterior função, permitindo criar um único ficheiro com todos os genes das várias condições analisadas [*Cufflinks.cuffmerge*], podendo o ficheiro resultante ser posteriormente utilizado para fazer a anotação dos genes ou isoformas com expressão. Este ficheiro pode ser também utilizado para fazer uma comparação [*Cufflinks.cuffcompare*] com um genoma já anotado, possibilitando a identificação de novos genes e efetuar unicamente a anotação dos mesmos. Por outro lado, este pode ser utilizado unicamente para calcular os genes ou isoformas com diferença significativa, no que diz respeito à sua expressão, utilizando-se a função [*Cufflinks.cuff diff*], que tem como ficheiros de entrada os ficheiros *BAM* calculados no *TopHat* e um ficheiro com a anotação no formato *GFF*, e tem como ficheiros de saída vários ficheiros *CSV* com dados importantes como o *p-value* e o "*fold change*" associados a cada gene ou isoforma.

CummeRbund: este programa, tal com o *DESeq2* encontra-se implementado em R. Uma vez que o *Cufflinks* não possibilita a visualização dos resultados e cálculo e exportação de imagens e gráficos a partir dos mesmos, foi desenvolvido o *CummeRbund* com o objetivo de colmatar essa lacuna. Este carrega os resultados do *CuffDiff* diretamente do local onde foram guardados e permite efetuar uma série de operações sob os mesmos. Uma vez que este se encontra implementado em R, é recomendado, tal como no caso do *DESeq2*, a sua utilização no *RStudio*. Devido ao elevado número de funcionalidades disponibilizadas pelo mesmo apenas serão referidas algumas, como por exemplo o carregamento dos dados [*CummeRbund.read _cufflinks*], o controlo de qualidade a partir da criação de um gráfico de dispersão [*CummeRbund.dispersion_plot*], a criação de uma matriz com a contagem das leituras normalizadas [*CummeRbund.count_matrix*], a seleção de um conjunto de genes [*CummeRbund.genes_set*]. A partir da seleção de genes é possível fazer um *heatmap* em relação à expressão dos mesmos com a função [*CummeRbund.heatmap*], ou tendo em conta um gene específico é possível identificar genes com comportamentos idênticos no que diz respeito à sua expressão [*CummeRbund.cs_cluster][CummeRbund.cs_cluster_plot*].

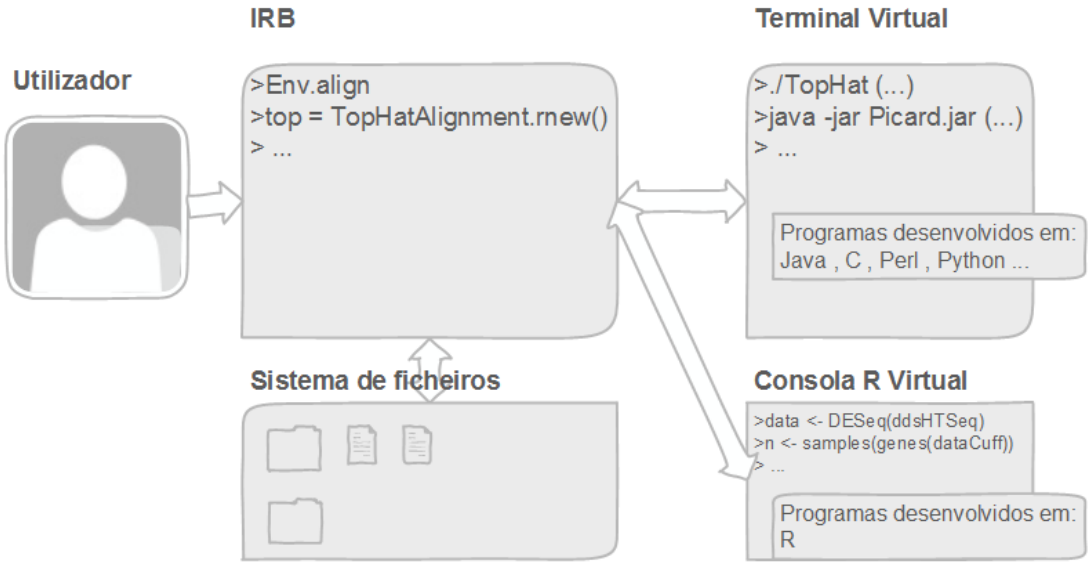


Figura 5-2 Visão geral das interações no sistema em estudos de RNA-Seq. O utilizador trabalha unicamente ao nível da consola Ruby, sendo as restantes operações invisíveis para o mesmo. O Env atribui os nomes e local aos novos ficheiros criados, e faz a leitura dos ficheiros necessários. É ainda responsabilidade do código em Ruby executar os programas, independentemente da linguagem de programação, ou a partir de uma sessão virtual de R, executar comandos em R e fazer a gestão dos objetos guardados na sessão.

O Env representativo de um estudo de RNA-Seq (Figura 5-2) segue os mesmos princípios de organização de módulos e funcionamento dos mesmos, utilizados para o DNA-Seq *de novo*, onde existe uma segmentação das várias funcionalidades em várias classes específicas representantes de um dado objetivo. Por exemplo, um "ambiente de trabalho" declarado pode ter vários alinhamentos associados, permitindo dessa forma a utilização de vários programas de alinhamento ou o processamento de vários alinhamentos com parâmetros diferentes, sendo possível a sua comparação, tanto ao nível de alinhamento das leituras como em relação aos genes onde a alteração na sua expressão foi considerada significativa. A grande vantagem é a partilha de parâmetros globais do ENV, como o genoma de referência e a anotação do mesmo, e a estrutura organizada do processamento dos dados. A partir de um dado alinhamento, é assim possível calcular a expressão diferencial dos genes, utilizando-se para isso o programa *DESeq2* ou o *Cufflinks*. É ainda possível identificar novos genes, sendo esta operação também ela feita ao nível do alinhamento.

5.3 Caso de estudo - Estudo da expressão genética da *Saccharomyces cerevisiae*

No caso de estudo que se segue será exemplificado o modo de funcionamento do sistema em estudo de RNA-Seq, sendo neste caso aplicado ao estudo da expressão genética da *S. cerevisiae* em duas condições distintas. Neste caso serão utilizados os comandos ao nível da

segunda camada (Ruby), sendo estes independentes de código implementado nas ferramentas da SilicoLife, o que não acontece no caso dos ENVs.

Será seguido o protocolo de análise e utilizados os dados referidos no artigo de Nookaew et al. [150]. Este artigo faz a comparação entre vários programas de alinhamento e cálculo de expressão genética de dados referentes a RNA-Seq, comparando ainda os resultados obtidos com os da tecnologia de microarrays. O organismo em estudo trata-se da *S. cerevisiae* CEN.PK 113-7D, estirpe muito utilizada em laboratórios. As amostras de *RNA* foram recolhidas em duas condições distintas, mais concretamente em estados de crescimento descontinuo (*batch)* e em crescimento contínuo (*chemostat* ou quimiostato). Estas foram recolhidas após ao estado estacionário ter sido atingido, tendo sido disponibilizadas 20 $g\ l^{-1}$ de glucose no caso do crescimento em *batch* e 10 $g\ l^{-1}$ no quimiostato de forma a manter um crescimento limitado pela fonte de carbono. As leituras com as identificações SRR453566, SRR453567, SRR453568, SRR453569, SRR453570 e SRR453571 foram retiradas da base de dados pública *SRA*, leituras estas correspondentes a três réplicas de amostras referentes às duas condições já descritas. Estas foram sequenciadas pela *Illumina Genome Analyzer* e são leituras *PE* com tamanho de 101 pb. Como referência foi utilizado o genoma da *S. cerevisiae* S288c, tal como a anotação do mesmo retirado da base de dados *SGD* (*Saccharomyces Genome Database*).

É importante referir no entanto que este caso de estudo não será uma comparação direta com o artigo citado, apesar de algumas comparações efetuadas no texto que se segue.

Análise**:** a análise iniciou-se com o *download* das leituras no formato *SRA* e a sua conversão para o formato *FASTQ*. A anotação do genoma foi também ela transferida da *SGD* e foram alterados os nomes dos cromossomas presentes no ficheiro *FASTA* de modo a serem coincidentes com os do ficheiro *GFF* referente à anotação.

O primeiro passo da análise foi o controlo de qualidade das leituras, tendo sido analisados os dados do relatório de qualidade das leituras calculado com a função do *FastQC* e a tendo em conta os mesmos foram utilizados os parâmetros mais indicados, como por exemplo o corte de 12 pb da extremidade esquerda da leitura.

```
>require 'ngs'
>NGS.qc('/SRR453566_1.fastq",ngs_reads2_path: "/SRR453566_2.fastq", threads: 8, left_trim_bases: 12, cut_off_qual_score: 25)
```

Comando 5-1 Primeiro comando permite carregar a gem desenvolvida. De seguida é feito controlo de qualidade das leituras cortando as bases de baixa qualidade da extremidade e eliminando as leituras que não tenham valores Phred superiores a 25 em pelo menos 70 % das bases.

Após o controlo de qualidade foram calculados dados relativos à distância entre as leituras e o desvio padrão desse valor. As leituras foram alinhadas com o programa *TopHat2* usando os parâmetros indicados para as leituras em análise.

```
>NGS::DNASeq::Bowtie2.build("S288C_ref.fasta")
>NGS::RNASeq::TopHat.align("SRR453566_1.fastq","/S288C_ref",ngs_reads2_path:
"SRR453566_2.fastq",gtf: "S288C.gff",mate_inner_dist: 200, mate_std_dev: 100, output_dir: "/batch/1")
```

Comando 5-2 Comando referente ao alinhamento das sequências de uma das réplicas. Inicialmente é construído o índice da sequência de referência, sendo de seguida feito o alinhamento das leituras. Uma vez que foi definido o parâmetro gtf, as leituras são inicialmente alinhadas tendo em conta o mesmo.

Tal como no artigo, foram marcadas e removidas as moléculas repetidas. Após o alinhamento e preparação dos ficheiros *BAM* foi feita a análise do transcriptoma utilizando-se dois programas distintos, o *DESeq2* e o *Cuffidff*. A pipeline onde é utilizado o *DESeq2* inicia-se pela utilização do programa *HTSeq*, e uma vez que este necessita obrigatoriamente de um ficheiro *SAM* de forma a processar os dados, todos os ficheiros *BAM* calculados foram convertidos para *SAM*. De seguida, a partir da matriz com os valores relativos às contagens das leituras é feito o estudo da expressão diferencial entre as duas condições.

```
>NGS::DNASeq::SamTools.bam_to_sam("/batch/1/batch_1.bam","/batch/1/batch_1.sam")
>NGS::RNASeq::HTSeq.count("/batch/1/batch_1.sam","/S288C.gff","/batch_1.counts",mode: "intersection-
strict", stranded: "no", featuretype: "gene", idattr: "ID")
>names = ['batch1','batch2',''batch3,'chemostat1','chemostat2','chemostat3']
>conditions = ['batch','batch',''batch,'chemostat','chemostat','chemostat']
>files = ['batch_1.counts',' batch_2.counts',' batch_3.counts','chemostat_1.counts',' 'chemostat_2.counts'',' 
'chemostat_3.counts'']
>levels = [''batch,'chemostat']
>NGS::RNASeq::DESeq2.differential_expression('/',files,names,conditions,levels,'deseq2')
```

Comando 5-3 Conversão dos ficheiros BAM para SAM. De seguida são calculadas as matrizes com a contagem das leituras e finalmente é feito o estudo relativo à expressão.

A utilização do *Cufflinks* para este efeito é mais simples, não existindo a necessidade de alterar os ficheiros resultantes do alinhamento.

```
>bam_files = [['/batch/1/batch_1.bam','/batch/2/batch_2.bam','/ batch/3/batch_3.bam'],['/ chemo-
stat/1/chemostat_1.bam','/ chemostat/2/chemostat_2.bam','/ chemostat/3/chemostat_3.bam']]
>NGS::RNASeq::Cufflinks.cuffdiff(''/S288C.gff,bam_files,output_dir: '/cuffdiff')
```

Comando 5-4 A partir dos ficheiros resultantes do alinhamento com o TopHat referentes às várias condições e réplicas de amostra é utilizado o Cuffdiff para o estudo da expressão dos genes.

Resultados: do ponto de vista do alinhamento das leituras os número e percentagem de leituras alinhadas contra a referência dada não são semelhantes aos apresentados no artigo, onde no caso das leituras referentes à condição *batch* foram obtidos valores médios na ordem dos 98%, e para as leituras da condição quimiostato valores médios próximos dos 95%. Neste caso de estudo os valores obtidos foram muito inferiores (Tabela 4-1), podendo existir várias razões para explicar os mesmos, por exemplo, a versão do *TopHat*, o programa utilizado para o controlo de qualidade que não tem o mesmo principio de funcionamento que o utilizado no sistema desenvolvido ou a utilização da anotação do genoma de referencia na altura do alinhamento, parâmetro não referido no artigo.

Tabela 5-1 Percentagem de leituras alinhadas. Leituras alinhas pelo programa TopHat em relação ao genoma e transcriptoma de referência.

Amostra	Número de Leituras	Percentagem Leituras Alinhadas
batch1	4382277	80,80%
batch2	6035456	85%
batch3	4261747	81,20%
quimiostato1	3081401	79,10%
quimiostato2	5177743	60,70%
quimiostato3	4832036	81,40%

Em relação ao número de genes onde existe uma alteração na sua expressão entre as duas condições, os valores não podem ser comparados, uma vez que a métrica de seleção dos mesmos é muito diferente. No artigo a métrica utilizada para fazer o corte foi o *q-value* < 10e-5, tendo sito utilizado neste caso de estudo selecionados os genes com *p-value* < 10e-4. Os valores obtidos para os duas alternativas implementadas obtiveram resultados bastante semelhantes (Figura 5-3), nomeadamente 3036 no caso da *pipeline* do *DESeq2* e 2856 no caso do *Cufflinks*.

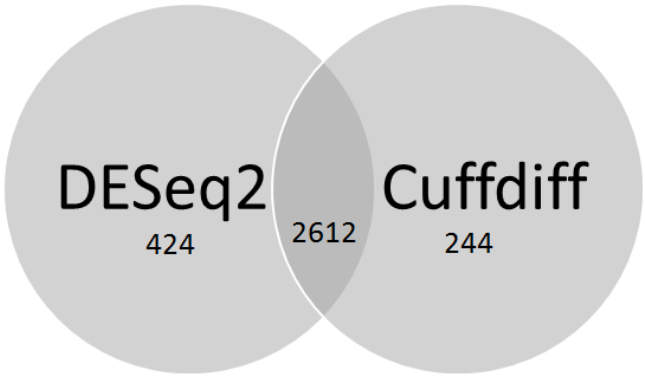


Figura 5-3 Número de genes com p-value inferior a 10e-4. Do conjunto de genes, 2612 foram obtidos por ambos os programas.

Por último, tal como no artigo, foram analisados os resultados ao nível biológico. A abordagem do enriquecimento de ontologias, neste caso com as identificações *GO* (*Gene Ontology)* permitem classificar e catalogar os genes segundo vários domínios, nomeadamente em relação ao componente celular, processo biológico e função molecular. A atribuição de identificações *GO* aos vários genes possibilita a verificação dos termos mais comuns no conjunto de genes com o *p-value* inferior ao definido, neste caso 10e-4, possibilitando posteriormente a identificação dos componentes, processos e funções que expliquem de certa forma o diferente comportamento biológico do organismo nas duas condições.

Para este cálculo foi utilizado o serviço *web Reporter Features* [162] para calcular os termos significantes utilizando para isso os genes com o *p-value* inferior ao referido. Os resultados obtidos (Figura 5-4), termos *GO* com *p-value* inferior a 10e-3, tal como os do artigo, explicam de forma correta o comportamento biológico associado às diferenças entre as condições de crescimento com excesso de glicose e limitado pela glicose. Pode ser corretamente identificado o grupo principal de termos *GO* (GO:0002181, GO:0042254, GO:0006364, GO:0000462, GO:00000027, GO:0006412, GO:0008152, GO:0055114) presentes no artigo, tanto na análise com o *DESeq2*, como com o *Cufflinks*.

Conclusão: apesar de ter perdido grande parte da simplicidade de execução do estudo intrínseca ao uso do ENV, esta forma de utilização do sistema permite que de uma forma independente ao código desenvolvido pela SilicoLife sejam feitas análises de dados RNA-Seq. Adquire-se igualmente uma maior flexibilidade em relação ao local e destino dos ficheiros processados, não sendo por isso necessário respeitar a organização do espaço de trabalho referida.

Em relação aos resultados obtidos será importante analisar a diferença nos valores referentes ao alinhamento das leituras quando comparados com os do artigo, e procurar perceber a razão dos mesmos. Quanto ao estudo dos genes verifica-se uma maior concordância em relação aos programas utilizados, havendo uma diferença muito baixa quando comparados com os valores referentes ao *Cufflinks* e *DESeq* apresentados no artigo.

O estudo dos termos *GO* permitiu identificar corretamente, tal como no artigo, termos diretamente relacionados com o crescimento celular, explicando de uma forma muito geral as diferenças entre o crescimento em *batch* e quimiostato. Verificou-se ainda uma maior sensibilidade por parte do *DESeq2* em relação ao *DESeq*.

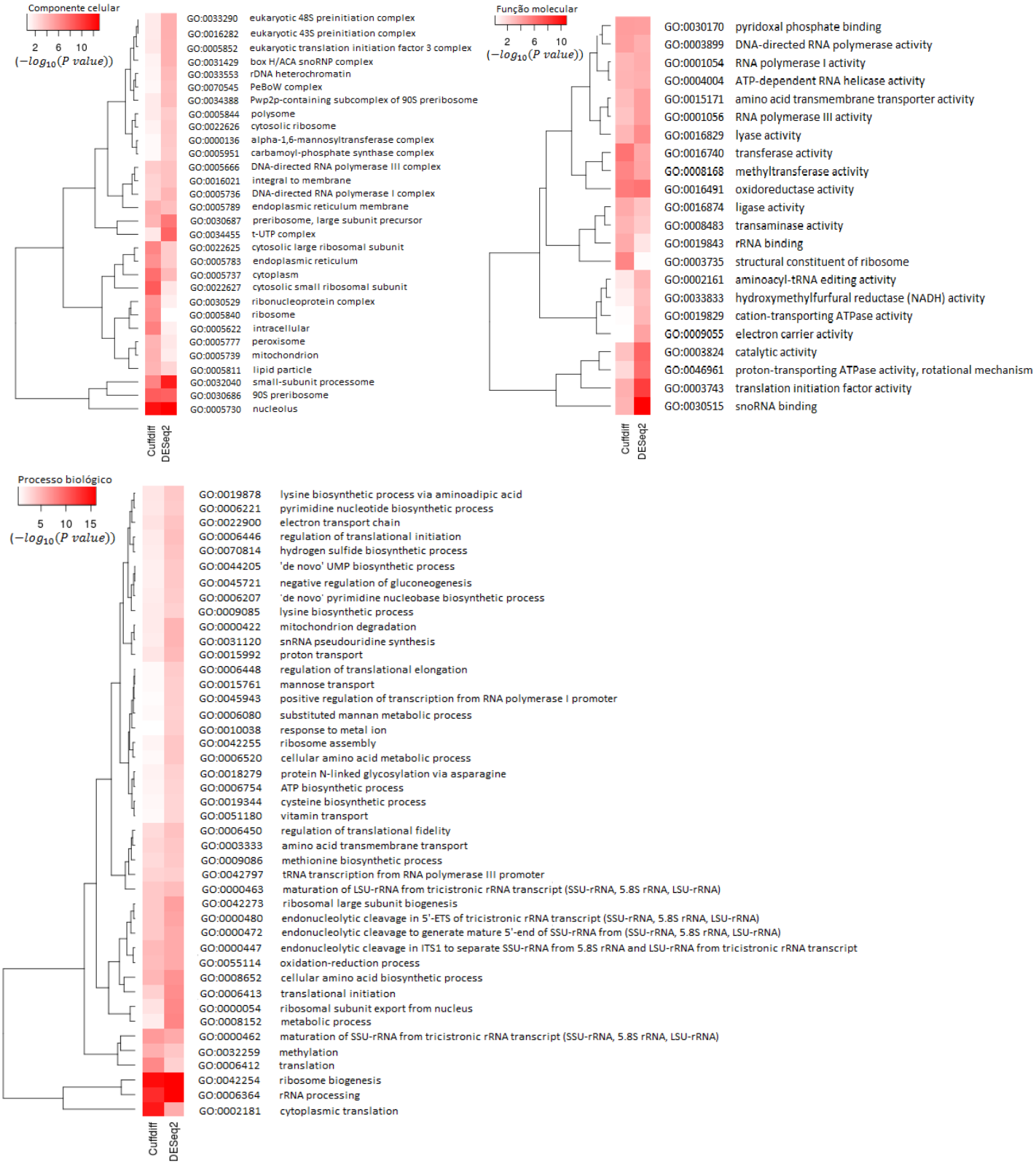


Figura 5-4 Heatmap dos termos GO com p-value inferior a 10e-4. Destacam-se termos como o nucleous, oxidoreductase activity e ribosome biogenesis, tendo sido calculados como significativos com ambos os programas implementados no sistema.

Capítulo 6

Conclusões

6.1 Resumo e principais contribuições

Neste trabalho, foi feito um levantamento do estado de arte dos programas desenvolvidos para processamento de dados de próxima geração de sequenciação. Foram descritos os princípios de funcionamento e funcionalidade principal de alguns dos programas referidos, tendo sido de seguida selecionados os programas que melhor correspondiam ao pretendido. Estes foram integrados no sistema de forma modular, abrindo assim possibilidade à fácil integração de novas ferramentas no futuro, bastando para isso que tenham em comum o mesmo tipo de ficheiros de entrada e saída. Foi ainda integrada nos módulos desenvolvidos documentação relacionada com a sua funcionalidade e com cada parâmetro das ferramentas.

O sistema implementado foi testado em três casos de estudo distintos. A sua simplicidade de execução foi mostrada no caso de estudo de DNA-Seq com referência, onde com a utilização de apenas alguns comandos, e sem a necessidade de saber quais os programas utilizados em todo o processo, foi capaz de gerar resultados que permitiram fazer uma comparação de duas estirpes de *Mycobacterium tuberculosis* e de as caracterizar tendo em conta os polimorfismos nelas encontrados.

Mostrou ser igualmente eficaz na assemblagem de um novo genoma, apresentando resultados muito parecidos em relação à quantidade de genes e números EC anotados nos *scaffolds* gerados. Os *scaffolds* gerados apresentaram ainda um alinhamento geral em relação ao genoma disponibilizado no NCBI de 99,65%, mostrando ser um bom ponto de início para a curação manual dos mesmos.

No caso de estudo relativo ao RNA-Seq ficou provada a concordância das ferramentas implementadas no que diz respeito à análise de expressão genérica, havendo 2600 genes onde foi considerada a existência de uma alteração significativa na sua expressão entre condições. A partir da análise de enriquecimento de ontologias mostrou-se ainda a capacidade

do sistema para encontrar significados biológicos que expliquem as diferenças entre as condições, conclusão chegada a partir da comparação dos termos identificados no caso de estudo com os do artigo mencionado no mesmo.

6.2 Trabalho futuro

6.2.1 Sistema integrado para o tratamento de dados de sequenciação de próxima geração

No geral, foram implementadas ferramentas com vista às principais funcionalidades necessárias. Uma vez que esta é uma área onde o desenvolvimento de novas ferramentas ainda é bastante frequente, haverá sempre espaço para a integração de novas ferramentas, caso apresentem vantagens em relação às já implementadas. Do ponto de vista geral, o próximo passo no desenvolvimento do sistema será o desenvolvimento de uma base de dados que o sustente, principalmente no que diz respeito à gestão de ficheiros considerados finais, como os relativos aos polimorfismos e ao nível de expressão dos vários genes, permitindo uma consulta simplificada dos dados por parte do utilizador.

Seria importante também o desenvolvimento de uma *interface* gráfica para o sistema. Esta seria utilizada para correr os vários programas, para definir *pipelines* personalizadas pelo utilizador e para a visualização dos dados guardados na base de dados.

Num futuro imediato, é ainda pretendida a disponibilização de um tutorial que mostre todos as possibilidades do sistema.

6.2.2 DNA-Seq com referência

Uma vez que é o módulo que se encontra desenvolvido há mais tempo, os principais objetivos são diretamente relacionados com esse facto:

- reformulação do ENV desenvolvido para estudos de DNA-Seq com referência;
- pesquisa e implementação de novos programas;
- implementação de alternativas ao *Samtools* para o cálculo de polimorfismos únicos.

6.2.3 DNA-Seq *de novo*

Os objetivos futuros para o módulo de DNA-Seq *de novo* passam por:

- Integração da ferramenta *MIRA* para a assemblagem de genomas;
- procurar perceber as discrepâncias dos resultados obtidos pelo Cisa em relação às outras ferramentas utilizadas. Fazer um estudo com vista a perceber em que casos será vantajoso a sua utilização.

6.2.4 RNA-seq

Em relação ao módulo de RNA-Seq os objetivos imediatos são:

- compreender porque razão os resultados do alinhamento verificados no caso de estudo da análise da expressão genética da *S. cerevisiae* não são equiparáveis aos apresentados no artigo;
- implementação de alternativas ao TopHat2 para o alinhamento de dados RNA-Seq.
- Integração de uma ferramenta para a análise de termos *GO*.

Referências Bibliográficas

[1] F. Sanger and S. Nicklen, "DNA sequencing with chain-terminating," *Biochemistry*, vol. 74, no. 12, pp. 5463–5467, 1977.

[2] L. Liu, Y. Li, S. Li, N. Hu, Y. He, R. Pong, D. Lin, L. Lu, and M. Law, "Comparison of next-generation sequencing systems.," *J. Biomed. Biotechnol.*, vol. 2012, p. 251364, Jan. 2012.

[3] J. Shendure and H. Ji, "Next-generation DNA sequencing.," *Nat. Biotechnol.*, vol. 26, no. 10, pp. 1135–45, Oct. 2008.

[4] G. S. Ginsburg and H. F. Willard, "Genomic and personalized medicine: foundations and applications.," *Transl. Res.*, vol. 154, no. 6, pp. 277–87, Dec. 2009.

[5] R. K. Varshney, S. N. Nayak, G. D. May, and S. A. Jackson, "Next-generation sequencing technologies and their implications for crop genetics and breeding.," *Trends Biotechnol.*, vol. 27, no. 9, pp. 522–30, Sep. 2009.

[6] "Integromics." [Online]. Disponível: https://www.integromics.com/. [Acedido: 09-Jul-2013].

[7] "Golden Helix | Genetic Association | SNP, CNV, NGS Analysis | Bioinformatics Services." [Online]. Disponível: http://www.goldenhelix.com/. [Acedido: 09-Jul-2013].

[8] "CLC bio - the world's leading bioinformatics analysis software - CLC bio." [Online]. Disponível: http://www.clcbio.com/. [Acedido: 09-Jul-2013].

[9] "SEQanswers Home." [Online]. Disponível: http://seqanswers.com/. [Acedido: 09-Jul-2013].

[10] "Bioinformatics Answers." [Online]. Disponível: http://www.biostars.org/. [Acedido: 09-Jul-2013].

[11] "SilicoLife :: Portugal." [Online]. Disponível: http://www.silicolife.pt/.

[12] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular Biology of the Cell*. Garland Science, 2007.

[13] "GA Pipeline - SamplePrepSlides.pdf." [Online]. Disponível: http://www.broadinstitute.org/files/shared/illuminavids/SamplePrepSlides.pdf. [Acedido: 02-Aug-2013].

[14] "Genome Spot: DNA library preparation for next-generation sequencing." [Online]. Disponível: http://genomespot.blogspot.pt/2012/10/dna-library-preparation-for-ngs.html. [Acedido: 06-Aug-2013].

[15] N. J. Croucher and N. R. Thomson, "Studying bacterial transcriptomes using RNA-seq.," *Curr. Opin. Microbiol.*, vol. 13, no. 5, pp. 619–24, Oct. 2010.

[16] J. Z. Levin, M. Yassour, X. Adiconis, C. Nusbaum, D. A. Thompson, N. Friedman, A. Gnirke, and A. Regev, "Comprehensive comparative analysis of strand-specific rnA sequencing methods," *Nat. Methods*, vol. 7, no. 9, 2010.

[17] "Next-Generation Sequencing | Life Technologies." [Online]. Disponível: http://www.invitrogen.com/site/us/en/home/Products-and-Services/Applications/Sequencing/Next-Generation-Sequencing.html. [Acedido: 09-Jul-2013].

[18] "Human genome race tweaked by Celera's stock offer - Forbes." [Online]. Disponível: http://www.forbes.com/1999/04/29/feat2.html. [Acedido: 09-Jul-2013].

[19] ": 454 Life Sciences, a Roche Company." [Online]. Disponível: http://www.454.com/. [Acedido: 09-Jul-2013].

[20] M. Margulies, M. Egholm, W. Altman, and S. Attiya, "Genome sequencing in microfabricated high-density picolitre reactors," *Nature*, vol. 437, no. 7057, pp. 376–380, 2005.

[21] "Illumina | sequencing and array-based solutions for genetic research." [Online]. Disponível: http://www.illumina.com/. [Acedido: 09-Jul-2013].

[22] E. R. Mardis, "Anticipating the 1,000 dollar genome.," *Genome Biol.*, vol. 7, no. 7, p. 112, Jan. 2006.

[23] "DNA Sequencing Costs." [Online]. Disponível: http://www.genome.gov/sequencingcosts/. [Acedido: 11-Jul-2013].

[24] E. R. Mardis, "Next-generation DNA sequencing methods.," *Annu. Rev. Genomics Hum. Genet.*, vol. 9, pp. 387–402, Jan. 2008.

[25] M. L. Metzker, "Sequencing technologies - the next generation.," *Nat. Rev. Genet.*, vol. 11, no. 1, pp. 31–46, Jan. 2010.

[26] "Ion Torrent." [Online]. Disponível: http://www.iontorrent.com/. [Acedido: 11-Jul-2013].

[27] "Pacific Biosciences: Home." [Online]. Disponível: http://www.pacificbiosciences.com/. [Acedido: 11-Jul-2013].

[28] "Oxford Nanopore Technologies." [Online]. Disponível: https://www.nanoporetech.com/. [Acedido: 11-Jul-2013].

[29] A. Comnish-bowden, "Nomenclature for incompletely specified bases in nucleic acid sequences: recommendations," vol. 13, no. 9, pp. 3021–3030, 1985.

[30] P. J. A. Cock, C. J. Fields, N. Goto, M. L. Heuer, and P. M. Rice, "The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants.," *Nucleic Acids Res.*, vol. 38, no. 6, pp. 1767–71, Apr. 2010.

[31] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, and R. Durbin, "The Sequence Alignment/Map format and SAMtools.," *Bioinformatics*, vol. 25, no. 16, pp. 2078–9, Aug. 2009.

[32] P. Danecek, A. Auton, G. Abecasis, C. a Albers, E. Banks, M. a DePristo, R. E. Handsaker, G. Lunter, G. T. Marth, S. T. Sherry, G. McVean, and R. Durbin, "The variant call format and VCFtools.," *Bioinformatics*, vol. 27, no. 15, pp. 2156–8, Aug. 2011.

[33] "A Hitchhiker's Guide to Next Generation Sequencing – Part 2 | Our 2 SNPs...®." [Online]. Disponível: http://blog.goldenhelix.com/?p=490. [Acedido: 11-Jul-2013].

[34] A. Oshlack, M. D. Robinson, and M. D. Young, "From RNA-seq reads to differential expression results.," *Genome Biol.*, vol. 11, no. 12, p. 220, Jan. 2010.

[35] R. Lister, B. D. Gregory, and J. R. Ecker, "Next is now: new technologies for sequencing of genomes, transcriptomes, and beyond.," *Curr. Opin. Plant Biol.*, vol. 12, no. 2, pp. 107–18, Apr. 2009.

[36] M. Dai, R. C. Thompson, C. Maher, R. Contreras-Galindo, M. H. Kaplan, D. M. Markovitz, G. Omenn, and F. Meng, "NGSQC: cross-platform quality analysis pipeline for deep sequencing data.," *BMC Genomics*, vol. 11 Suppl 4, no. Suppl 4, p. S7, Jan. 2010.

[37] S. Pabinger, A. Dander, M. Fischer, R. Snajder, M. Sperk, M. Efremova, B. Krabichler, M. R. Speicher, J. Zschocke, Z. Trajanoski, and T. Zlatko, "A survey of tools for variant analysis of next-generation genome sequencing data.," *Brief. Bioinform.*, Jan. 2013.

[38] P. C. Dolan and D. R. Denver, "TileQC: a system for tile-based quality control of Solexa data.," *BMC Bioinformatics*, vol. 9, p. 250, Jan. 2008.

[39] a Martínez-Alcántara, E. Ballesteros, C. Feng, M. Rojas, H. Koshinsky, V. Y. Fofanov, P. Havlak, and Y. Fofanov, "PIQA: pipeline for Illumina G1 genome analyzer data quality assessment.," *Bioinformatics*, vol. 25, no. 18, pp. 2438–9, Sep. 2009.

[40] "Babraham Bioinformatics - FastQC A Quality Control tool for High Throughput Sequence Data." [Online]. Disponível: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/. [Acedido: 19-Jul-2013].

[41] R. K. Patel and M. Jain, "NGS QC Toolkit: a toolkit for quality control of next generation sequencing data.," *PLoS One*, vol. 7, no. 2, p. e30619, Jan. 2012.

[42] "FASTX-Toolkit." [Online]. Disponível: http://hannonlab.cshl.edu/fastx_toolkit/. [Acedido: 19-Jul-2013].

[43] S. F. Altschul, T. L. Madden, Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.," *Nucleic Acids Res.*, vol. 25, no. 17, pp. 3389–402, Sep. 1997.

[44] H. Li and N. Homer, "A survey of sequence alignment algorithms for next-generation sequencing.," *Brief. Bioinform.*, vol. 11, no. 5, pp. 473–83, Sep. 2010.

[45] P. Flicek and E. Birney, "Sense from sequence reads : methods for alignment and assembly," vol. 6, no. 11, 2010.

[46] R. Li, C. Yu, Y. Li, T.-W. Lam, S.-M. Yiu, K. Kristiansen, and J. Wang, "SOAP2: an improved ultrafast tool for short read alignment.," *Bioinformatics*, vol. 25, no. 15, pp. 1966–7, Aug. 2009.

[47] H. Li, J. Ruan, and R. Durbin, "Mapping short DNA sequencing reads and calling variants using mapping quality scores," *Genome Res.*, pp. 1851–1858, 2008.

[48] T. D. Wu and C. K. Watanabe, "GMAP: a genomic mapping and alignment program for mRNA and EST sequences.," *Bioinformatics*, vol. 21, no. 9, pp. 1859–75, May 2005.

[49] Z. Ning, A. Cox, and J. Mullikin, "SSAHA: a fast search method for large DNA databases," *Genome Res.*, no. 2, pp. 1725–1729, 2001.

[50] B. Langmead, C. Trapnell, M. Pop, and S. L. Salzberg, "Ultrafast and memory-efficient alignment of short DNA sequences to the human genome.," *Genome Biol.*, vol. 10, no. 3, p. R25, Jan. 2009.

[51] B. Langmead and S. L. Salzberg, "Fast gapped-read alignment with Bowtie 2.," *Nat. Methods*, vol. 9, no. 4, pp. 357–9, Apr. 2012.

[52] H. Li and R. Durbin, "Fast and accurate short read alignment with Burrows-Wheeler transform.," *Bioinformatics*, vol. 25, no. 14, pp. 1754–60, Jul. 2009.

[53] N. a Fonseca, J. Rung, A. Brazma, and J. C. Marioni, "Tools for mapping high-throughput sequencing data.," *Bioinformatics*, vol. 28, no. 24, pp. 3169–3177, Oct. 2012.

[54] R. K. Aziz, D. Bartels, A. Best, M. DeJongh, T. Disz, R. a Edwards, K. Formsma, S. Gerdes, E. M. Glass, M. Kubal, F. Meyer, G. J. Olsen, R. Olson, A. L. Osterman, R. a Overbeek, L. K. McNeil, D. Paarmann, T. Paczian, B. Parrello, G. D. Pusch, C. Reich, R. Stevens, O. Vassieva, V. Vonstein, A. Wilke, and O. Zagnitko, "The RAST Server: rapid annotations using subsystems technology.," *BMC Genomics*, vol. 9, p. 75, Jan. 2008.

[55] A. Mckenna, M. Hanna, E. Banks, A. Sivachenko, K. Cibulskis, A. Kernytsky, K. Garimella, D. Altshuler, S. Gabriel, M. Daly, and M. A. Depristo, "The Genome Analysis Toolkit : A MapReduce framework for analyzing next-generation DNA sequencing data," pp. 1297–1303, 2010.

[56] R. Li, Y. Li, X. Fang, H. Yang, J. J. Wang, and K. Kristiansen, "SNP detection for massively parallel whole-genome resequencing.," *Genome Res.*, vol. 19, no. 6, pp. 1124–32, Jun. 2009.

[57] D. F. Simola and J. Kim, "Sniper: improved SNP discovery by multiply mapping deep sequenced reads.," *Genome Biol.*, vol. 12, no. 6, p. R55, Jan. 2011.

[58] R. Nielsen, J. S. Paul, A. Albrechtsen, and Y. S. Song, "Genotype and SNP calling from next-generation sequencing data.," *Nat. Rev. Genet.*, vol. 12, no. 6, pp. 443–51, Jun. 2011.

[59] "VCF (Variant Call Format) version 4.1 | 1000 Genomes." [Online]. Disponível: http://www.1000genomes.org/wiki/Analysis/Variant Call Format/vcf-variant-call-format-version-41. [Acedido: 16-Jul-2013].

[60] P. Cingolani, V. M. Patel, M. Coon, T. Nguyen, S. J. Land, D. M. Ruden, and X. Lu, "Using Drosophila melanogaster as a Model for Genotoxic Chemical Mutational Studies with a New Program, SnpSift.," *Front. Genet.*, vol. 3, no. March, p. 35, Jan. 2012.

[61] P. Cingolani, A. Platts, M. Coon, and T. Nguyen, "A program for annotating and predicting the effects of single nucleotide polymorphisms, SnpEff: SNPs in the genome of Drosophila melanogaster strain w1118; iso-2; iso," *Fly.*, no. June, pp. 1–13, 2012.

[62] K. Wang, M. Li, and H. Hakonarson, "ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data.," *Nucleic Acids Res.*, vol. 38, no. 16, p. e164, Sep. 2010.

[63] W. McLaren, B. Pritchard, D. Rios, Y. Chen, P. Flicek, and F. Cunningham, "Deriving the consequences of genomic variants with the Ensembl API and SNP Effect Predictor.," *Bioinformatics*, vol. 26, no. 16, pp. 2069–70, Aug. 2010.

[64] A. Magi, M. Benelli, A. Gozzini, F. Girolami, F. Torricelli, and M. L. Brandi, "Bioinformatics for Next Generation Sequencing Data," *Genes.*, vol. 1, no. 2, pp. 294–307, Sep. 2010.

[65] H. Bao, H. Guo, J. Wang, R. Zhou, X. Lu, and S. Shi, "MapView: visualization of short reads alignment on a desktop computer.," *Bioinformatics*, vol. 25, no. 12, pp. 1554–5, Jun. 2009.

[66] I. Milne, M. Bayer, L. Cardle, P. Shaw, G. Stephen, F. Wright, and D. Marshall, "Tablet–next generation sequence assembly visualization.," *Bioinformatics*, vol. 26, no. 3, pp. 401–2, Feb. 2010.

[67] J. T. Robinson, H. Thorvaldsdóttir, W. Winckler, M. Guttman, E. S. Lander, G. Getz, and J. P. Mesirov, "Integrative genomics viewer.," *Nat. Biotechnol.*, vol. 29, no. 1, pp. 24–6, Jan. 2011.

[68] A. Darling, B. Mau, F. Blattner, and N. Perna, "Mauve: multiple alignment of conserved genomic sequence with rearrangements," *Genome Res.*, pp. 1394–1403, 2004.

[69] M. Baker, "De novo genome assembly: what every biologist should know," *Nat. Methods*, vol. 9, no. 4, pp. 333–337, Mar. 2012.

[70] J. R. Miller, S. Koren, and G. Sutton, "Assembly algorithms for next-generation sequencing data.," *Genomics*, vol. 95, no. 6, pp. 315–27, Jun. 2010.

[71] R. L. Warren, G. G. Sutton, S. J. M. Jones, and R. a Holt, "Assembling millions of short DNA sequences using SSAKE.," *Bioinformatics*, vol. 23, no. 4, pp. 500–1, Feb. 2007.

[72] "Unidade 2 - Assemblagem de Genomas." [Online]. Disponível: http://www.dcc.fc.up.pt/~pribeiro/aulas/bioinformatica1213/unidade2.pdf. [Acedido: 17-Jul-2013].

[73] "Products - Analysis Software : 454 Life Sciences, a Roche Company." [Online]. Disponível: http://454.com/products/analysis-software/index.asp. [Acedido: 19-Jul-2013].

[74] D. Hernandez, P. François, L. Farinelli, M. Osterås, and J. Schrenzel, "De novo bacterial genome sequencing: millions of very short reads assembled on a desktop computer.," *Genome Res.*, vol. 18, no. 5, pp. 802–9, May 2008.

[75] J. R. Miller, A. L. Delcher, S. Koren, E. Venter, B. P. Walenz, A. Brownley, J. Johnson, K. Li, C. Mobarry, and G. Sutton, "Aggressive assembly of pyrosequencing reads with mates.," *Bioinformatics*, vol. 24, no. 24, pp. 2818–24, Dec. 2008.

[76] D. R. Zerbino and E. Birney, "Velvet: algorithms for de novo short read assembly using de Bruijn graphs.," *Genome Res.*, vol. 18, no. 5, pp. 821–9, May 2008.

[77] J. T. Simpson, K. Wong, S. D. Jackman, J. E. Schein, S. J. M. Jones, and I. Birol, "ABySS: a parallel assembler for short read sequence data.," *Genome Res.*, vol. 19, no. 6, pp. 1117–23, Jun. 2009.

[78] R. Li, H. Zhu, J. Ruan, W. Qian, X. Fang, Z. Shi, Y. Li, S. S. Li, G. Shan, K. Kristiansen, H. Yang, and J. J. Wang, "De novo assembly of human genomes with massively parallel short read sequencing.," *Genome Res.*, vol. 20, no. 2, pp. 265–72, Feb. 2010.

[79] W. Zhang, J. Chen, Y. Yang, Y. Tang, J. Shang, and B. Shen, "A practical comparison of de novo genome assembly software tools for next-generation sequencing technologies.," *PLoS One*, vol. 6, no. 3, p. e17915, Jan. 2011.

[80] B. Schmidt, R. Sinha, B. Beresford-Smith, and S. J. Puglisi, "A fast hybrid short read fragment assembly algorithm.," *Bioinformatics*, vol. 25, no. 17, pp. 2279–80, Sep. 2009.

[81] "De Novo Assembly Using Illumina Reads - technote_denovo_assembly_ecoli.pdf." [Online]. Disponível: http://res.illumina.com/documents/products/technotes/technote_denovo_assembly_ecoli.pdf. [Acedido: 19-Jul-2013].

[82] B. Chevreux, "MIRA: an automated genome and EST assembler," *Ruprecht-Karls Univ.*, 2005.

[83] M. Pop, D. S. Kosack, and S. L. Salzberg, "Hierarchical Scaffolding With Bambus," *Genome Res.*, no. 301, pp. 149–159, 2004.

[84] A. Dayarian, T. P. Michael, and A. M. Sengupta, "SOPRA: Scaffolding algorithm for paired reads via statistical optimization.," *BMC Bioinformatics*, vol. 11, p. 345, Jan. 2010.

[85] M. Boetzer, C. V Henkel, H. J. Jansen, D. Butler, and W. Pirovano, "Scaffolding pre-assembled contigs using SSPACE.," *Bioinformatics*, vol. 27, no. 4, pp. 578–9, Feb. 2011.

[86] E. D. Green, "Strategies for the systematic sequencing of complex genomes.," *Nat. Rev. Genet.*, vol. 2, no. 8, pp. 573–83, Aug. 2001.

[87] M. Boetzer and W. Pirovano, "Toward almost closed genomes with GapFiller.," *Genome Biol.*, vol. 13, no. 6, p. R56, Jan. 2012.

[88] R. Luo, B. Liu, Y. Xie, Z. Li, W. Huang, J. Yuan, G. He, Y. Chen, Q. Pan, Y. Liu, J. Tang, G. Wu, H. Zhang, Y. Shi, Y. Liu, C. Yu, B. Wang, Y. Lu, C. Han, D. W. Cheung, S.-M. Yiu, S. Peng, Z. Xiaoqian, G. Liu, X. Liao, Y. Li, H. Yang, J. Wang, T.-W. Lam, and J. Wang, "SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler.," *Gigascience*, vol. 1, no. 1, p. 18, Jan. 2012.

[89] I. J. Tsai, T. D. Otto, and M. Berriman, "Improving draft assemblies by iterative mapping and assembly of short reads to eliminate gaps.," *Genome Biol.*, vol. 11, no. 4, p. R41, Jan. 2010.

[90] S. Kurtz, A. Phillippy, A. L. Delcher, M. Smoot, M. Shumway, C. Antonescu, and S. L. Salzberg, "Versatile and open software for comparing large genomes.," *Genome Biol.*, vol. 5, no. 2, p. R12, Jan. 2004.

[91] S.-H. Lin and Y.-C. Liao, "CISA: contig integrator for sequence assembly of bacterial genomes.," *PLoS One*, vol. 8, no. 3, p. e60843, Jan. 2013.

[92] G. Yao, L. Ye, H. Gao, P. Minx, W. C. Warren, and G. M. Weinstock, "Graph accordance of next-generation sequence assemblies.," *Bioinformatics*, vol. 28, no. 1, pp. 13–6, Jan. 2012.

[93] a L. Delcher, D. Harmon, S. Kasif, O. White, and S. L. Salzberg, "Improved microbial gene identification with GLIMMER.," *Nucleic Acids Res.*, vol. 27, no. 23, pp. 4636–41, Dec. 1999.

[94] D. Hyatt, G.-L. Chen, P. F. Locascio, M. L. Land, F. W. Larimer, and L. J. Hauser, "Prodigal: prokaryotic gene recognition and translation initiation site identification.," *BMC Bioinformatics*, vol. 11, p. 119, Jan. 2010.

[95] C. Burge and S. Karlin, "Prediction of complete gene structures in human genomic DNA.," *J. Mol. Biol.*, vol. 268, no. 1, pp. 78–94, Apr. 1997.

[96] "geneid homepage." [Online]. Disponível: http://genome.crg.es/software/geneid/.

[97] J. Besemer and M. Borodovsky, "GeneMark: web software for gene finding in prokaryotes, eukaryotes and viruses.," *Nucleic Acids Res.*, vol. 33, no. Web Server issue, pp. W451–4, Jul. 2005.

[98] R. D. Finn, J. Clements, and S. R. Eddy, "HMMER web server: interactive sequence similarity searching.," *Nucleic Acids Res.*, vol. 39, no. Web Server issue, pp. W29–37, Jul. 2011.

[99] A. Conesa, S. Götz, J. M. García-Gómez, J. Terol, M. Talón, and M. Robles, "Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research.," *Bioinformatics*, vol. 21, no. 18, pp. 3674–6, Sep. 2005.

[100] a Schulze and J. Downward, "Navigating gene expression using microarrays–a technology review.," *Nat. Cell Biol.*, vol. 3, no. 8, pp. E190–5, Aug. 2001.

[101] V. Velculescu and L. Zhang, "Serial analysis of gene expression," *Science (80-.).*, 1995.

[102] S. Brenner, M. Johnson, J. Bridgham, G. Golda, D. H. Lloyd, D. Johnson, S. Luo, S. McCurdy, M. Foy, M. Ewan, R. Roth, D. George, S. Eletr, G. Albrecht, E. Vermaas, S. R. Williams, K. Moon, T. Burcham, M. Pallas, R. B. DuBridge, J. Kirchner, K. Fearon, J. Mao, and K. Corcoran, "Gene expression analysis by massively parallel signature sequencing (MPSS) on microbead arrays.," *Nat. Biotechnol.*, vol. 18, no. 6, pp. 630–4, Jun. 2000.

[103] S. Marguerat and J. Bähler, "RNA-seq: from technology to biology.," *Cell. Mol. Life Sci.*, vol. 67, no. 4, pp. 569–79, Feb. 2010.

[104] G. R. Grant, M. H. Farkas, A. D. Pizarro, N. F. Lahens, J. Schug, B. P. Brunk, C. J. Stoeckert, J. B. Hogenesch, and E. a Pierce, "Comparative analysis of RNA-Seq alignment algorithms and the RNA-Seq unified mapper (RUM).," *Bioinformatics*, vol. 27, no. 18, pp. 2518–28, Sep. 2011.

[105] C. Trapnell, L. Pachter, and S. L. Salzberg, "TopHat: discovering splice junctions with RNA-Seq.," *Bioinformatics*, vol. 25, no. 9, pp. 1105–11, May 2009.

[106] D. Kim, G. Pertea, C. Trapnell, H. Pimentel, R. Kelley, and S. L. Salzberg, "TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions.," *Genome Biol.*, vol. 14, no. 4, p. R36, Apr. 2013.

[107] K. Wang, D. Singh, Z. Zeng, S. J. Coleman, Y. Huang, G. L. Savich, X. He, P. Mieczkowski, S. a Grimm, C. M. Perou, J. N. Macleod, D. Y. Chiang, J. F. Prins, and J. Liu, "MapSplice: accurate mapping of RNA-seq reads for splice junction discovery.," *Nucleic Acids Res.*, vol. 38, no. 18, p. e178, Oct. 2010.

[108] K. F. Au, H. Jiang, L. Lin, Y. Xing, and W. H. Wong, "Detection of splice junctions from paired-end RNA-seq data by SpliceMap.," *Nucleic Acids Res.*, vol. 38, no. 14, pp. 4570–8, Aug. 2010.

[109] "Survey: RNA-Seq analysis for Differential Gene/Transcript Expression - SEQanswers." [Online]. Disponível: http://seqanswers.com/forums/showthread.php?p=90423. [Acedido: 19-Jul-2013].

[110] Q.-Y. Zhao, Y. Wang, Y.-M. Kong, D. Luo, X. Li, and P. Hao, "Optimizing de novo transcriptome assembly from short-read RNA-Seq data: a comparative study.," *BMC Bioinformatics*, vol. 12 Suppl 1, no. Suppl 14, p. S2, Jan. 2011.

[111] M. G. Grabherr, B. J. Haas, M. Yassour, J. Z. Levin, D. a Thompson, I. Amit, X. Adiconis, L. Fan, R. Raychowdhury, Q. Zeng, Z. Chen, E. Mauceli, N. Hacohen, A. Gnirke, N. Rhind, F. di Palma, B. W. Birren, C. Nusbaum, K. Lindblad-Toh, N. Friedman, and A. Regev, "Full-length transcriptome assembly from RNA-Seq data without a reference genome.," *Nat. Biotechnol.*, vol. 29, no. 7, pp. 644–52, Jul. 2011.

[112] M. H. Schulz, D. R. Zerbino, M. Vingron, and E. Birney, "Oases: robust de novo RNA-seq assembly across the dynamic range of expression levels.," *Bioinformatics*, vol. 28, no. 8, pp. 1086–92, Apr. 2012.

[113] I. Birol, S. D. Jackman, C. B. Nielsen, J. Q. Qian, R. Varhol, G. Stazyk, R. D. Morin, Y. Zhao, M. Hirst, J. E. Schein, D. E. Horsman, J. M. Connors, R. D. Gascoyne, M. a Marra, and S. J. M. Jones, "De novo transcriptome assembly with ABySS.," *Bioinformatics*, vol. 25, no. 21, pp. 2872–7, Nov. 2009.

[114] J. a Martin and Z. Wang, "Next-generation transcriptome assembly.," *Nat. Rev. Genet.*, vol. 12, no. 10, pp. 671–82, Oct. 2011.

[115] M. Garber, M. G. Grabherr, M. Guttman, and C. Trapnell, "Computational methods for transcriptome annotation and quantification using RNA-seq.," *Nat. Methods*, vol. 8, no. 6, pp. 469–77, Jun. 2011.

[116] G. Chen, C. Wang, and T. Shi, "Overview of Disponível methods for diverse RNA-Seq data analyses.," *Sci. China. Life Sci.*, vol. 54, no. 12, pp. 1121–8, Dec. 2011.

[117] M. Griffith, O. Griffith, and J. Mwenifumbo, "Alternative expression analysis by RNA sequencing," *Nature*, vol. 7, no. 10, 2010.

[118] C. Trapnell, A. Roberts, L. Goff, G. Pertea, D. Kim, D. R. Kelley, H. Pimentel, S. L. Salzberg, J. L. Rinn, and L. Pachter, "Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks.," *Nat. Protoc.*, vol. 7, no. 3, pp. 562–78, Mar. 2012.

[119] "HTSeq: Analysing high-throughput sequencing data with Python — HTSeq v0.5.4p2 documentation." [Online]. Disponível: http://www-huber.embl.de/users/anders/HTSeq/doc/overview.html. [Acedido: 23-Jul-2013].

[120] M.-A. Dillies, A. Rau, J. Aubert, C. Hennequet-Antier, M. Jeanmougin, N. Servant, C. Keime, G. Marot, D. Castel, J. Estelle, G. Guernec, B. Jagla, L. Jouneau, D. Laloë, C. Le Gall, B. Schaëffer, S. Le Crom, M. Guedj, and F. Jaffrézic, "A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis.," *Brief. Bioinform.*, Sep. 2012.

[121] M. D. Robinson, D. J. McCarthy, and G. K. Smyth, "edgeR: a Bioconductor package for differential expression analysis of digital gene expression data.," *Bioinformatics*, vol. 26, no. 1, pp. 139–40, Jan. 2010.

[122] S. Anders and W. Huber, "Differential expression analysis for sequence count data.," *Genome Biol.*, vol. 11, no. 10, p. R106, Jan. 2010.

[123] L. Wang, Z. Feng, X. Wang, and X. Zhang, "DEGseq: an R package for identifying differentially expressed genes from RNA-seq data.," *Bioinformatics*, vol. 26, no. 1, pp. 136–8, Jan. 2010.

[124] G. Dennis, B. T. Sherman, D. a Hosack, J. Yang, W. Gao, H. C. Lane, and R. a Lempicki, "DAVID: Database for Annotation, Visualization, and Integrated Discovery.," *Genome Biol.*, vol. 4, no. 5, p. P3, Jan. 2003.

[125] X. Zhou and Z. Su, "EasyGO: Gene Ontology-based annotation and functional enrichment analysis tool for agronomical species.," *BMC Genomics*, vol. 8, p. 246, Jan. 2007.

[126] M. D. Young, M. J. Wakefield, G. K. Smyth, and A. Oshlack, "Gene ontology analysis for RNA-seq: accounting for selection bias.," *Genome Biol.*, vol. 11, no. 2, p. R14, Jan. 2010.

[127] B. Giardine, C. Riemer, R. C. Hardison, R. Burhans, P. Shah, Y. Zhang, D. Blankenberg, I. Albert, W. Miller, W. J. Kent, and A. Nekrutenko, "Galaxy : A platform for interactive large-scale genome analysis," pp. 1451–1455, 2005.

[128] A. Prlić, A. Yates, S. E. Bliven, P. W. Rose, J. Jacobsen, P. V Troshin, M. Chapman, J. Gao, C. H. Koh, S. Foisy, R. Holland, G. Rimsa, M. L. Heuer, H. Brandstätter-Müller, P. E. Bourne, S. Willis, H. Brandsta, and G. Rims, "BioJava: an open-source framework for bioinformatics in 2012.," *Bioinformatics*, vol. 28, no. 20, pp. 2693–5, Oct. 2012.

[129] J. E. Stajich, D. Block, K. Boulez, S. E. Brenner, S. A. Chervitz, C. Dagdigian, G. Fuellen, J. G. R. Gilbert, I. Korf, H. Lapp, H. Lehva, C. Matsalla, C. J. Mungall, B. I. Osborne, M. R. Pocock, P. Schattner, M. Senger, L. D. Stein, E. Stupka, M. D. Wilkinson, and E. Birney, "The Bioperl Toolkit : Perl Modules for the Life Sciences," pp. 1611–1618, 2002.

[130] R. C. Gentleman, V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, F. Leisch, C. Li, M. Maechler, A. J. Rossini, G. Sawitzki, C. Smith, G. Smyth, L. Tierney, J. Y. H. Yang, and J. Zhang, "Bioconductor: open software development for computational biology and bioinformatics.," *Genome Biol.*, vol. 5, no. 10, p. R80, Jan. 2004.

[131] N. Goto, P. Prins, M. Nakao, R. Bonnal, J. Aerts, and T. Katayama, "BioRuby: bioinformatics software for the Ruby programming language.," *Bioinformatics*, vol. 26, no. 20, pp. 2617–9, Oct. 2010.

[132] T. Barrett, S. E. Wilhite, P. Ledoux, C. Evangelista, I. F. Kim, M. Tomashevsky, K. a Marshall, K. H. Phillippy, P. M. Sherman, M. Holko, A. Yefanov, H. Lee, N. Zhang, C. L. Robertson, N. Serova, S. Davis, and A. Soboleva, "NCBI GEO: archive for functional genomics data sets–update.," *Nucleic Acids Res.*, vol. 41, no. D1, pp. D991–5, Jan. 2013.

[133] G. Rustici, N. Kolesnikov, M. Brandizi, T. Burdett, M. Dylag, I. Emam, A. Farne, E. Hastings, J. Ison, M. Keays, N. Kurbatova, J. Malone, R. Mani, A. Mupo, R. Pedro Pereira, E. Pilicheva, J. Rung, A. Sharma, Y. A. Tang, T. Ternent, A. Tikhonov, D. Welter, E. Williams, A. Brazma, H. Parkinson, U. Sarkans, and R. P. Pereira, "ArrayExpress update--trends in database growth and links to data analysis tools.," *Nucleic Acids Res.*, vol. 41, no. Database issue, pp. D987–90, Jan. 2013.

[134] Y. Kodama, J. Mashima, E. Kaminuma, T. Gojobori, O. Ogasawara, T. Takagi, K. Okubo, and Y. Nakamura, "The DNA Data Bank of Japan launches a new resource, the DDBJ Omics Archive of functional genomics experiments.," *Nucleic Acids Res.*, vol. 40, no. Database issue, pp. D38–42, Jan. 2012.

[135] R. Leinonen, H. Sugawara, and M. Shumway, "The sequence read archive.," *Nucleic Acids Res.*, vol. 39, no. Database issue, pp. D19–21, Jan. 2011.

[136] E. T. Cirulli and D. B. Goldstein, "Uncovering the roles of rare variants in common disease through whole-genome sequencing.," *Nat. Rev. Genet.*, vol. 11, no. 6, pp. 415–25, Jun. 2010.

[137] DePristo, "A framework for variation discovery and genotyping using next-generation DNA sequencing data," *Nat. ...*, vol. 43, no. 5, pp. 491–498, 2011.

[138] M. Meyerson, S. Gabriel, and G. Getz, "Advances in understanding cancer genomes through second-generation sequencing.," *Nat. Rev. Genet.*, vol. 11, no. 10, pp. 685–96, Oct. 2010.

[139] J. Bras, R. Guerreiro, and J. Hardy, "Use of next-generation sequencing and other whole-genome strategies to dissect neurological disease.," *Nat. Rev. Neurosci.*, vol. 13, no. 7, pp. 453–64, Jul. 2012.

[140] D. MacLean, J. D. G. Jones, and D. J. Studholme, "Application of 'next-generation' sequencing technologies to microbial genetics," *Nat. Rev. Microbiol.*, vol. 7, no. April, pp. 287–296, Feb. 2009.

[141] E. R. Mardis, "A decade's perspective on DNA sequencing technology.," *Nature*, vol. 470, no. 7333, pp. 198–203, Feb. 2011.

[142] X. Fu, N. Fu, S. Guo, Z. Yan, Y. Xu, H. Hu, C. Menzel, W. Chen, Y. Li, R. Zeng, and P. Khaitovich, "Estimating accuracy of RNA-Seq and microarrays with proteomics.," *BMC Genomics*, vol. 10, no. 1, p. 161, Jan. 2009.

[143] J. H. Malone and B. Oliver, "Microarrays, deep sequencing and the true measure of the transcriptome.," *BMC Biol.*, vol. 9, p. 34, Jan. 2011.

[144] N. Raghavachari, J. Barb, Y. Yang, P. Liu, K. Woodhouse, D. Levy, C. J. O'Donnell, P. J. Munson, and G. J. Kato, "A systematic comparison and evaluation of high density exon arrays and RNA-seq technology used to unravel the peripheral blood transcriptome of sickle cell disease.," *BMC Med. Genomics*, vol. 5, no. 1, p. 28, Jan. 2012.

[145] J. R. Bradford, Y. Hey, T. Yates, Y. Li, S. D. Pepper, and C. J. Miller, "A comparison of massively parallel nucleotide sequencing with oligonucleotide microarrays for global transcription profiling.," *BMC Genomics*, vol. 11, no. 1, p. 282, Jan. 2010.

[146] B. Friedman and T. Maniatis, "ExpressionPlot: a web-based framework for analysis of RNA-Seq and microarray gene expression data.," *Genome Biol.*, vol. 12, no. 7, p. R69, Jan. 2011.

[147] F. Ozsolak and P. M. Milos, "RNA sequencing: advances, challenges and opportunities.," *Nat. Rev. Genet.*, vol. 12, no. 2, pp. 87–98, Feb. 2011.

[148] A. Goncalves, A. Tikhonov, A. Brazma, and M. Kapushesky, "A pipeline for RNA-seq data processing and quality assessment.," *Bioinformatics*, vol. 27, no. 6, pp. 867–9, Mar. 2011.

[149] A. Mortazavi, B. A. Williams, K. Mccue, L. Schaeffer, and B. Wold, "Mapping and quantifying mammalian transcriptomes by RNA-Seq," *Nat. Methods*, vol. 5, no. 7, 2008.

[150] I. Nookaew, M. Papini, N. Pornputtpong, G. Scalcinati, L. Fagerberg, M. Uhlén, and J. Nielsen, "A comprehensive comparison of RNA-Seq-based transcriptome analysis from reads to differential gene expression and cross-comparison with microarrays: a case study in Saccharomyces cerevisiae.," *Nucleic Acids Res.*, pp. 1–14, Sep. 2012.

[151] N. R. Hackett, M. W. Butler, R. Shaykhiev, J. Salit, L. Omberg, J. L. Rodriguez-Flores, J. G. Mezey, Y. Strulovici-Barel, G. Wang, L. Didon, and R. G. Crystal, "RNA-Seq quantification of the human small airway epithelium transcriptome.," *BMC Genomics*, vol. 13, no. 1, p. 82, Jan. 2012.

[152] C. M. Gowen and S. S. Fong, "Genome-scale metabolic model integrated with RNAseq data to identify metabolic states of Clostridium thermocellum.," *Biotechnol. J.*, vol. 5, no. 7, pp. 759–67, Jul. 2010.

[153] P. a Jensen and J. a Papin, "Functional integration of a metabolic network model and expression data without arbitrary thresholding.," *Bioinformatics*, vol. 27, no. 4, pp. 541–7, Feb. 2011.

[154] D. Lee, K. Smallbone, W. B. Dunn, E. Murabito, C. L. Winder, and D. B. Kell, "Improving metabolic flux predictions using absolute gene expression data," *Manchester 1824*, 2012.

[155] S. M. Brown, *Next-Generation DNA Sequencing informatics*. Cold Spring Harbor Laboratory Press, 2013.

[156] A. Sandgren, M. Strong, P. Muthukrishnan, B. K. Weiner, G. M. Church, and M. B. Murray, "Tuberculosis drug resistance mutation database.," *PLoS Med.*, vol. 6, no. 2, p. e2, Feb. 2009.

[157] H. Zhang, D. Li, L. Zhao, J. Fleming, N. Lin, T. Wang, Z. Liu, C. Li, N. Galwey, J. Deng, Y. Zhou, Y. Zhu, Y. Gao, T. Wang, S. Wang, Y. Huang, M. Wang, Q. Zhong, L. Zhou, T. Chen, J. Zhou, R. Yang, G. Zhu, H. Hang, J. Zhang, F. Li, K. Wan, J. Wang, X.-E. Zhang, and L. Bi, "Genome sequencing of 161 Mycobacterium tuberculosis isolates from China identifies genes and intergenic regions associated with drug resistance.," *Nat. Genet.*, vol. 45, no. 10, pp. 1255–60, Oct. 2013.

[158] P. Soares-Castro and P. Santos, "Towards the description of the genome catalogue of Pseudomonas sp. strain M1," *Genome Announc.*, vol. 1, no. 1, pp. 11–12, 2013.

[159] K. Malde, "Flower: extracting information from pyrosequencing data.," *Bioinformatics*, vol. 27, no. 7, pp. 1041–2, Apr. 2011.

[160] "VBC | Victorian Bioinformatics Consortium | Prokka." [Online]. Disponível: http://www.vicbioinformatics.com/software.prokka.shtml. [Acedido: 24-Oct-2013].

[161] Y. Moriya, M. Itoh, S. Okuda, A. C. Yoshizawa, and M. Kanehisa, "KAAS: an automatic genome annotation and pathway reconstruction server.," *Nucleic Acids Res.*, vol. 35, no. Web Server issue, pp. W182–5, Jul. 2007.

[162] "ReportFeatures-Toolbox - Nielsen's Lab for Systems Biology." [Online]. Disponível: http://129.16.106.142/toolbox.php#reporterfeatures. [Acedido: 29-Oct-2013].

Anexo A - Visão geral de um estudo de DNA-Seq com referência

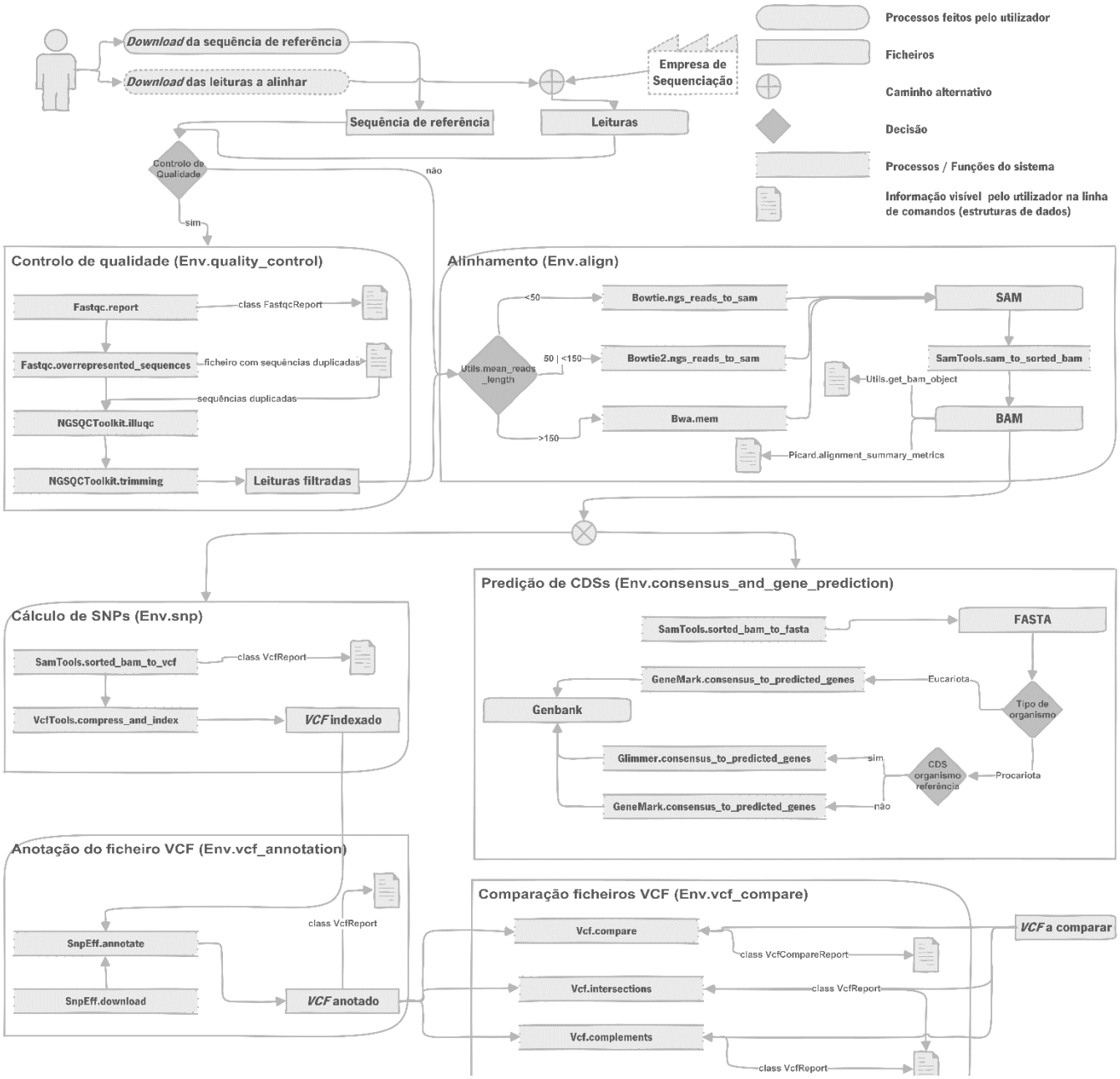


Figura A-1 Visão geral de uma análise tipo de um estudo de DNA-Seq com referência.

Anexo B - Genomas de referência *Mycobacterium tuberculosis*

Tabela B-1 Genomas utilizados como referência

Estirpe	NC	Tamanho	Ncbi	Genes	Proteínas
CCDC5079_uid161943	NC_017523	4398812	link	3695	3646
CCDC5180_uid161941	NC_017522	4405981	link	3638	3590
CDC1551_uid57775	NC_002755	4403837	link	4293	4189
CTRI_2_uid161997	NC_017524	4398525	link	4001	3944
F11_uid58417	NC_009565	4424435	link	3998	3941
H37Ra_uid58853	NC_009525	4419977	link	4084	4034
H37Rv_uid170532	NC_018143	4411708	link	4170	4111
H37Rv_uid57777	NC_000962	4411532	link	4111	4018
KZN_1435_uid59069	NC_012943	4398250	link	4107	4059
KZN_4207_uid83619	NC_016768	4394985	link	4044	3996
KZN_605_uid54947	NC_018078	4399120	link	4071	4001
RGTB327_uid157907	NC_017026	4380119	link	3739	3691
RGTB423_uid162179	NC_017528	4406587	link	3670	3622
UT205_uid162179	NC_016934	4418088	link	3814	3796

Anexo C - Mapas *KEGG*

Tabela C-2 Tabela com os mapas KEGG gerados com diferenças (8) em relação ao número de proteínas anotadas presentes no mapa. O total de mapas gerados foi 189.

	Celera	*NCBI*
00500 Starch and sucrose metabolism	18	17
00330 Arginine and proline metabolism	53	54
00550 Peptidoglycan biosynthesis	18	17
00860 Porphyrin and chlorophyll metabolism	35	36
00130 Ubiquinone and other terpenoid-quinone biosynthesis	11	10
00625 Chloroalkane and chloroalkene degradation	12	11
00791 Atrazine degradation	6	5
03010 Ribosome	53	54